



Panduan Pengguna

AWS CloudFormation



Versi API 2010-05-15

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

AWS CloudFormation: Panduan Pengguna

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

Merek dagang dan tampilan dagang Amazon tidak boleh digunakan sehubungan dengan produk atau layanan apa pun yang bukan milik Amazon, dengan cara apa pun yang dapat menyebabkan kebingungan di antara pelanggan, atau dengan cara apa pun yang menghina atau mendiskreditkan Amazon. Semua merek dagang lain yang tidak dimiliki oleh Amazon merupakan hak milik masing-masing pemiliknya, yang mungkin atau tidak terafiliasi, terkait dengan, atau disponsori oleh Amazon.

Table of Contents

Apa itu AWS CloudFormation?	1
Menyederhanakan manajemen infrastruktur	1
Mereplikasi infrastruktur Anda dengan cepat	1
Mengendalikan dan melacak perubahan pada infrastruktur Anda dengan mudah	2
Memulai dengan CloudFormation	2
Informasi terkait	2
Memulai	4
Bagaimana cara CloudFormation kerja	4
Konsep utama	4
Bagaimana cara CloudFormation kerja	9
Cara untuk memulai CloudFormation	12
Mendaftar untuk Akun AWS	12
Mendaftar untuk Akun AWS	12
Buat pengguna dengan akses administratif	13
Membuat tumpukan pertama Anda	14
Prasyarat	15
Buat CloudFormation tumpukan dengan konsol	15
Pantau pembuatan tumpukan	19
Uji server web	20
Pemecahan Masalah	21
Bersihkan	22
Langkah selanjutnya	23
Praktik terbaik	24
Mempersingkat loop umpan balik untuk meningkatkan kecepatan pengiriman	25
Atur tumpukan Anda berdasarkan siklus hidup dan kepemilikan	25
Gunakan referensi lintas tumpukan untuk mengeksport sumber daya bersama	26
Verifikasi kuota untuk semua jenis sumber daya	27
Gunakan kembali templat untuk mereplikasi tumpukan di beberapa lingkungan	27
Gunakan modul untuk menggunakan kembali konfigurasi sumber daya	27
Gunakan AWS tipe parameter -spesifik	28
Gunakan batasan parameter	28
Gunakan parameter semu untuk mempromosikan portabilitas	29
Gunakan <code>AWS::CloudFormation::Init</code> untuk menyebarkan aplikasi perangkat lunak di instans Amazon EC2	30

Gunakan skrip pembantu terbaru	30
Validasi templat sebelum digunakan	30
Validasi template untuk kepatuhan kebijakan organisasi	31
Kelola semua sumber daya tumpukan melalui CloudFormation	31
Buat rangkaian perubahan sebelum memperbarui tumpukan	32
Gunakan kebijakan tumpukan	32
Gunakan ulasan kode dan kontrol revisi untuk mengelola templat	32
Perbarui EC2 instans Amazon Anda secara teratur	33
Bekerja dengan templat	34
Tempat template disimpan	35
Memvalidasi template	35
Memulai dengan template	2
Rencanakan untuk menggunakan referensi CloudFormation template	36
Sampel template	37
Format templat	37
Struktur templat	38
Komentar	40
Spesifikasi	41
Pelajari selengkapnya	41
Menggunakan ekspresi reguler	41
Bagian templat	42
Resources	44
Parameters	53
Outputs	67
Mappings	72
Metadata	80
Rules	86
Conditions	106
Transform	114
Versi format	115
Description	116
Komposer Infrastruktur	117
Mengapa menggunakan Infrastructure Composer dalam mode CloudFormation konsol?	117
Bagaimana mode ini berbeda dari konsol Infrastructure Composer?	117
Generator IAC	118
Pertimbangan	119

Izin	120
Perintah yang umum digunakan	120
Migrasikan template ke AWS CDK	121
Mulai pemindaian sumber daya	121
Lihat ringkasan pemindaian	126
Buat template dari sumber daya yang dipindai	127
Buat tumpukan dari sumber daya yang dipindai	132
Selesaikan properti hanya tulis	133
Dapatkan nilai yang disimpan di layanan lain	139
Pertimbangan umum	139
Dapatkan nilai plaintext Systems Manager	140
Dapatkan string aman Systems Manager	143
Dapatkan nilai Secrets Manager	146
Dapatkan AWS nilai	151
Sintaksis	151
Parameter pseudo yang tersedia	152
Contoh	153
Dapatkan output tumpukan	157
Mengekspor nilai output tumpukan versus menggunakan tumpukan bersarang	158
Pertimbangan	159
Mencantumkan nilai output yang diekspor	159
Pencantuman tumpukan yang mengimpor nilai output yang diekspor	160
Tentukan sumber daya yang ada saat runtime	161
Gambaran Umum	162
Contoh	163
Pertimbangan	166
Jenis parameter AWS spesifik yang didukung	167
Jenis parameter Systems Manager yang Didukung	169
Tipe parameter Systems Manager yang tidak didukung	169
Panduan	170
Lihat output sumber daya di tumpukan lain CloudFormation	171
Bergabung dengan VPC di akun lain	176
Buat aplikasi yang diskalakan dan seimbang beban	186
Menyebarkan aplikasi di Amazon EC2	200
Memperbarui tumpukan	210
Lakukan penyebaran ECS blue/green	245

Cuplikan templat	269
Umum	270
Penskalaan otomatis	280
AWS Konsol Penagihan	313
AWS CloudFormation	317
CloudFront	324
CloudWatch	336
CloudWatch Log	342
DynamoDB	381
Amazon EC2	387
Amazon ECS	425
Amazon EFS	448
Elastic Beanstalk	473
Penyeimbang Beban Elastis	478
IAM	480
AWS Lambda	498
AWS OpsWorks	507
Amazon Redshift	516
Amazon RDS	526
Route 53	534
Amazon S3	543
Amazon SNS	551
Amazon SQS	552
Amazon Timestream	553
Windowstumpukan berbasis	582
Tumpukan bootstrap Windows	583
Gunakan jenis sumber daya yang CloudFormation disediakan	590
Sumber daya khusus	591
Makro templat	645
Tumpukan nested	663
Kondisi tunggu	670
Buat konfigurasi sumber daya yang dapat digunakan kembali dengan modul	676
Pertimbangan saat menggunakan modul	678
Memahami pembuatan versi modul	678
Gunakan modul dalam template	680
Gunakan parameter untuk menentukan nilai modul	681

Sumber daya modul referensi	684
Membuat dan mengelola tumpukan	686
Antarmuka untuk mengelola tumpukan Anda	688
Buat tumpukan	688
Membuat tumpukan	689
Konfigurasikan opsi tumpukan	692
Pratinjau konfigurasi tumpukan Anda	694
Lihat informasi tumpukan	696
Perbarui template tumpukan Anda	698
Memahami perilaku pembaruan sumber daya tumpukan	700
Perbarui tumpukan menggunakan set perubahan	702
Buat set perubahan	704
Melihat set perubahan	710
Jalankan set perubahan	718
Hapus set perubahan	721
Contoh set perubahan	722
Set perubahan untuk tumpukan yang di-nest	735
Perbarui tumpukan secara langsung	742
Batalkan pembaruan tumpukan	745
Untuk membatalkan pembaruan tumpukan (konsol)	746
Untuk membatalkan pembaruan tumpukan (AWS CLI)	746
Hapus tumpukan	746
Sumber daya terkait	748
Lihat tumpukan yang dihapus	748
Pantau kemajuan tumpukan	749
Lihat acara tumpukan	749
Lihat grafik timeline penyebaran tumpukan	757
Memahami peristiwa pembuatan tumpukan	760
Pantau pembaruan tumpukan	762
Terus mengembalikan pembaruan	764
Tentukan penyebab kegagalan tumpukan	768
Opsi kegagalan tumpukan	770
Gulung kembali tumpukan Anda pada pelanggaran alarm	780
Tambahkan pemicu rollback selama pembuatan tumpukan atau pembaruan	781
Tambahkan pemicu rollback ke set perubahan	782
Lihat pemicu rollback untuk tumpukan	782

Lihat pemicu rollback untuk set perubahan	782
Mendeteksi perubahan konfigurasi yang tidak terkelola pada tumpukan dan sumber daya	783
Apa yang dimaksud dengan penyimpangan?	783
Kode status deteksi penyimpangan	785
Pertimbangan saat mendeteksi penyimpangan	787
Mendeteksi penyimpangan di seluruh tumpukan CloudFormation	790
Mendeteksi penyimpangan di sumber daya tumpukan individu	795
Mengatasi penyimpangan dengan operasi impor	799
Impor AWS sumber daya	809
Impor AWS sumber daya secara manual	810
Mengimpor AWS sumber daya secara otomatis	849
Pengembalian operasi impor	850
Stack refactoring	856
Cara kerja stack refactoring	856
Batasan refactoring tumpukan	857
Memfaktorkan ulang tumpukan menggunakan AWS Command Line Interface	860
Dukungan jenis sumber daya	862
Gunakan tautan buat cepat untuk membuat tumpukan CloudFormation	1057
Format URL	1057
Contoh	1059
Membuat tumpukan menggunakan tautan buat cepat	1059
Contoh perintah untuk AWS CLI dan PowerShell	1060
CancelUpdateStack	1061
ContinueUpdateRollback	1062
CreateStack	1063
Deploy	1068
DeleteStack	1068
DescribeStackEvents	1070
DescribeStackResource	1071
DescribeStackResources	1073
DescribeStacks	1076
GetTemplate	1080
ListStackResources	1081
ListStacks	1083
UpdateStack	1085
ValidateTemplate	1089

Unggah artefak lokal ke bucket S3	1091
Mengelola tumpukan dengan StackSets	1095
StackSets konsep	1096
Administrator dan akun target	1097
AWS CloudFormation StackSets	1097
Model izin untuk StackSets	1097
Instans tumpukan	1098
StackSet operasi	1099
StackSet opsi operasi	1100
Tanda	1102
StackSets kode status	1103
Kode status instance tumpukan	1104
Prasyarat	1105
Aktifkan Wilayah AWS yang dinonaktifkan secara default	1106
Memberikan izin yang dikelola sendiri	1107
Aktifkan akses terpercaya	1123
Mulai menggunakan template sampel	1128
Prasyarat	1129
Buat StackSet dengan contoh template dari konsol	1129
Pantau StackSet pembuatan	1132
Lihat StackSet hasil	1132
Perbarui StackSet	1133
Tambahkan tumpukan ke StackSet	1133
Bersihkan	1134
Langkah selanjutnya	1136
Buat StackSets (izin yang dikelola sendiri)	1136
Buat StackSet dengan izin yang dikelola sendiri (konsol)	1137
Buat StackSet dengan izin yang dikelola sendiri (AWS CLI)	1139
Buat StackSets (izin yang dikelola layanan)	1140
Pertimbangan	1141
Buat StackSet dengan izin yang dikelola layanan (konsol)	1142
Buat StackSet dengan izin yang dikelola layanan (AWS CLI)	1144
Aktifkan-nonaktifkan penerapan otomatis	1149
Cara kerja penerapan otomatis	1149
Pertimbangan	1149
Mengaktifkan atau menonaktifkan penerapan otomatis (konsol)	1150

Mengaktifkan atau menonaktifkan penerapan otomatis (AWS CLI)	1150
Perbarui StackSets	1151
Perbarui StackSet (konsol)	1151
Perbarui StackSet (AWS CLI)	1153
Tambahkan tumpukan ke StackSets	1155
Tambahkan tumpukan ke StackSet (konsol)	1155
Tambahkan tumpukan ke StackSet (AWS CLI)	1157
Ganti parameter pada tumpukan	1158
Ganti parameter pada tumpukan (konsol)	1159
Ganti parameter pada tumpukan (AWS CLI)	1161
Hapus tumpukan dari StackSets	1162
Hapus tumpukan dari StackSet (konsol)	1162
Hapus tumpukan dari StackSet (AWS CLI)	1164
Hapus StackSets	1166
Hapus StackSet (konsol)	1166
Hapus a StackSet (AWS CLI)	1166
Hapus peran layanan (opsional)	1167
Gerbang akun target	1168
Persyaratan	1168
CloudFormation template untuk membuat fungsi Lambda	1169
Pilih Mode Konkurensi	1169
Cara kerja setiap Mode Konkurensi	1170
Memilih antara toleransi kegagalan yang ketat dan toleransi kegagalan lunak berdasarkan kecepatan penerapan	1173
Memilih Mode Konkurensi (konsol)	1175
Memilih Mode Konkurensi Anda (AWS CLI)	1175
Mendeteksi drift pada StackSets	1176
Bagaimana CloudFormation melakukan deteksi drift pada StackSet	1177
Mendeteksi penyimpangan pada StackSet (konsol)	1178
Mendeteksi penyimpangan pada StackSet (AWS CLI)	1179
Menghentikan deteksi drift pada StackSet	1186
Impor tumpukan ke StackSets	1187
Impor tumpukan yang dikelola sendiri	1187
Impor tumpukan yang dikelola layanan	1191
Kembalikan impor tumpukan	1196
Praktik terbaik	1196

Penentuan templat	1197
Membuat atau menambahkan tumpukan ke StackSet	1197
Memperbarui tumpukan di StackSet	1198
Sampel template	1198
Pemecahan Masalah	1200
Alasan umum untuk kegagalan operasi tumpukan	1200
Mencoba lagi pembuatan tumpukan atau operasi pembaruan yang gagal	1201
Penghapusan instans tumpukan gagal	1202
Operasi impor tumpukan gagal	1202
Jumlah kegagalan instance stack untuk StackSets operasi	1202
Menyinkronkan tumpukan dengan kode sumber Git	1209
Cara kerja sinkronisasi Git	1210
Cara kerja sinkronisasi Git	1210
Komentar pada permintaan tarik	1212
File penyebaran tumpukan	1212
CloudFormation berkas templat	1213
Repositori definisi templat	1213
Prasyarat	1214
Repositori Git	1214
CloudFormation Template	1214
Peran IAM	1215
Izin yang diperlukan	1215
Kebijakan kepercayaan	1217
Buat tumpukan dari kode sumber repositori	1217
Buat tumpukan dari kode sumber repositori	1217
Perbarui tumpukan Anda dari repositori Git Anda	1220
Aktifkan komentar pada permintaan tarik	1220
Dasbor status	1221
Status sinkronisasi Git	1221
Acara sinkronisasi terbaru	1222
Status tumpukan yang didukung	1223
Mengelola ekstensi dengan CloudFormation registri	1227
Dokumentasi terkait	1227
Konsep	1228
Jenis ekstensi	1228
Ekstensi publik	1229

Ekstensi yang diaktifkan	1229
Lihat ekstensi yang tersedia dan diaktifkan	1230
Ekstensi publik	1231
Peran IAM	1232
Secara otomatis menggunakan versi ekstensi baru	1234
Gunakan alias untuk merujuk ke ekstensi	1235
AWS CLI Perintah yang umum digunakan untuk bekerja dengan ekstensi publik	1235
Aktifkan ekstensi publik	1236
Perbarui ekstensi publik	1240
Nonaktifkan ekstensi publik	1241
Ekstensi pribadi	1243
Izin IAM	1243
AWS CLI Perintah yang umum digunakan untuk bekerja dengan ekstensi pribadi	1243
Daftarkan ekstensi pribadi	1244
Deregister ekstensi pribadi	1246
Mengedit data konfigurasi untuk ekstensi	1247
Izin diperlukan untuk menggunakan referensi dinamis	1247
Mengedit data konfigurasi untuk ekstensi (konsol)	1248
Mengedit data konfigurasi untuk ekstensi (AWS CLI)	1249
Mencatat jenis sumber daya di AWS Config	1250
Mencegah properti sensitif dicatat dalam item konfigurasi	1251
Pengiriman berkelanjutan	1252
Panduan: Membangun alur untuk tumpukan uji dan produksi	1252
Prasyarat	1253
Gambaran umum panduan	1253
Langkah 1: Edit artifact dan unggah ke Bucket S3	1254
Langkah 2: Buat tumpukan alur	1256
Langkah 3: Lihat WordPress tumpukan	1261
Langkah 4: Bersihkan Sumber Daya	1262
Lihat juga	1263
Referensi properti konfigurasi	1263
Properti konfigurasi (konsol)	1264
Properti konfigurasi (objek JSON)	1267
Lihat juga	1271
AWS CloudFormation artefak	1272
File templat tumpukan	1272

File konfigurasi templat	1273
Lihat juga	1274
Menggunakan fungsi penggantian parameter dengan saluran pipa CodePipeline	1274
Fn::GetArtifactAtt	1275
Fn::GetParam	1276
Lihat juga	1279
Panduan Referensi Template	1280
Keamanan	1281
Lindungi tumpukan agar tidak dihapus	1282
Mengendalikan siapa yang dapat mengubah perlindungan terminasi pada tumpukan	1284
Mencegah pembaruan ke sumber daya tumpukan	1284
Contoh kebijakan tumpukan	1285
Menetapkan kebijakan tumpukan	1286
Mengatur kebijakan tumpukan	1290
Memperbarui sumber daya dilindungi	1292
Mengubah kebijakan tumpukan	1295
Lebih banyak contoh kebijakan tumpukan	1296
Perlindungan data	1299
Enkripsi diam	1300
Enkripsi bergerak	1301
Privasi lalu lintas jaringan internet	1301
Kontrol akses dengan IAM	1301
Mendefinisikan kebijakan berbasis identitas IAM untuk CloudFormation	1302
Mengakui sumber daya IAM dalam templat CloudFormation	1311
Mengelola kredensi untuk aplikasi yang berjalan di instans Amazon EC2	1312
Memberikan akses sementara (akses federasi)	1312
Contoh kebijakan berbasis identitas	1314
AWS CloudFormation peran layanan	1321
Pencegahan "confused deputy" lintas layanan	1322
Permintaan FAS dan evaluasi izin	1324
Mencatat panggilan API	1326
CloudFormation informasi di CloudTrail	1327
Memahami entri file CloudFormation log	1328
Keamanan infrastruktur	1332
Ketahanan	1333
Validasi kepatuhan	1333

Konfigurasi dan analisis kerentanan	1334
Praktik terbaik keamanan	1334
Gunakan IAM untuk mengontrol akses	1335
Jangan menanamkan kredensial dalam templat Anda	1335
Gunakan AWS CloudTrail untuk mencatat CloudFormation panggilan	1336
AWS PrivateLink	1336
Pertimbangan untuk titik akhir CloudFormation VPC	1336
Buat VPC endpoint antarmuka untuk CloudFormation	1338
Membuat kebijakan VPC endpoint untuk CloudFormation	1338
Pemantauan EventBridge dengan	1340
CloudFormation dan ikhtisar acara sinkronisasi Git	1341
Izin	1343
Membuat pola acara khusus	1343
Referensi detail acara	1344
Perubahan Status Sumber Daya	1345
Perubahan Status Stack	1348
Perubahan Status Deteksi Drift	1350
StackSet Perubahan Status	1352
StackSet Perubahan Status Instans Stack	1354
StackSetPerubahan Status Operasi	1356
Perubahan Status Sinkronisasi Repositori	1358
Perubahan Status Sinkronisasi Sumber Daya	1361
Kuota	1364
Ketersediaan fitur	1373
StackSets dan makro	1373
Pemecahan Masalah	1374
Panduan pemecahan masalah	1374
Memecahkan masalah kesalahan	1375
Gagal menghapus tumpukan	1375
Kesalahan dependensi	1376
AWS Config dan AWS Systems Manager konflik	1377
Kesalahan penguraian parameter saat melewati daftar	1377
Izin IAM tidak mencukupi	1378
Nilai tidak valid atau properti sumber daya tidak didukung	1378
Kuota terlampaui	1378
Kegagalan rollback tumpukan bersarang	1379

Tidak ada pembaruan untuk dilakukan	1379
Sumber daya gagal distabilkan selama membuat, memperbarui, atau menghapus operasi tumpukan	1380
Grup keamanan tidak ada di VPC	1380
Pembaruan rollback gagal	1381
Kondisi tunggu tidak menerima jumlah sinyal yang diperlukan dari EC2 instans Amazon ...	1382
Sumber daya dihilangkan dari tumpukan tetapi tidak dihapus	1383
Menghubungi dukungan	1384
Pemecahan masalah dengan Amazon Q	1385
Fitur	1385
Cara kerjanya	1386
Harga	1386
Ketersediaan wilayah	1386
Batasan	1386
Riwayat dokumen	1387
Pembaruan yang diarsipkan	1416
.....	mdxxxiii

Apa itu AWS CloudFormation?

AWS CloudFormation adalah layanan yang membantu Anda memodelkan dan mengatur AWS sumber daya Anda sehingga Anda dapat menghabiskan lebih sedikit waktu mengelola sumber daya tersebut dan lebih banyak waktu untuk berfokus pada aplikasi Anda yang berjalan AWS. Anda membuat template yang menjelaskan semua AWS sumber daya yang Anda inginkan (seperti instans Amazon atau EC2 instans Amazon RDS DB), dan CloudFormation menangani penyediaan dan konfigurasi sumber daya tersebut untuk Anda. Anda tidak perlu membuat dan mengonfigurasi AWS sumber daya secara individual dan mencari tahu apa yang bergantung pada apa; CloudFormation menangani itu. Skenario berikut menunjukkan bagaimana CloudFormation dapat membantu.

Menyederhanakan manajemen infrastruktur

Untuk aplikasi web yang dapat diskalakan yang juga menyertakan database backend, Anda dapat menggunakan grup Auto Scaling, penyeimbang beban Elastic Load Balancing, dan instance database Amazon Relational Database Service. Anda dapat menggunakan setiap layanan individu untuk menyediakan sumber daya ini dan setelah Anda membuat sumber daya, Anda harus mengonfigurasinya untuk bekerja sama. Semua tugas ini dapat menambah kerumitan dan waktu bahkan sebelum Anda mengaktifkan dan menjalankan aplikasi.

Sebagai gantinya, Anda dapat membuat CloudFormation template atau memodifikasi yang sudah ada. Template menjelaskan semua sumber daya Anda dan propertinya. Saat Anda menggunakan template tersebut untuk membuat CloudFormation tumpukan, berikan CloudFormation grup Auto Scaling, load balancer, dan database untuk Anda. Setelah tumpukan berhasil dibuat, AWS sumber daya Anda aktif dan berjalan. Anda dapat menghapus tumpukan dengan mudah, yang menghapus semua sumber daya dalam tumpukan. Dengan menggunakan CloudFormation, Anda dengan mudah mengelola kumpulan sumber daya sebagai satu unit.

Mereplikasi infrastruktur Anda dengan cepat

Jika aplikasi Anda memerlukan ketersediaan tambahan, Anda dapat mereplikasinya di beberapa wilayah sehingga jika satu wilayah tidak tersedia, pengguna Anda masih dapat menggunakan aplikasi Anda di wilayah lain. Tantangan dalam mereplikasi aplikasi Anda adalah bahwa hal itu juga mengharuskan Anda untuk mereplikasi sumber daya Anda. Anda tidak hanya perlu merekam semua sumber daya yang diperlukan aplikasi Anda, tetapi Anda juga harus menyediakan dan mengonfigurasi sumber daya tersebut di setiap wilayah.

Gunakan kembali CloudFormation template Anda untuk membuat sumber daya Anda secara konsisten dan berulang. Untuk menggunakan kembali templat Anda, jelaskan sumber daya Anda satu kali, lalu sediakan sumber daya yang sama berulang kali di beberapa wilayah.

Mengendalikan dan melacak perubahan pada infrastruktur Anda dengan mudah

Dalam beberapa kasus, Anda mungkin memiliki sumber daya dasar yang ingin Anda tingkatkan secara bertahap. Misalnya, Anda dapat mengubah ke jenis instans berkinerja lebih tinggi dalam konfigurasi peluncuran Auto Scaling sehingga Anda dapat mengurangi jumlah maksimum instans dalam grup Auto Scaling Anda. Jika masalah terjadi setelah Anda menyelesaikan pembaruan, Anda mungkin perlu mengembalikan infrastruktur Anda ke pengaturan awal. Untuk melakukannya secara manual, Anda tidak hanya harus mengingat sumber daya mana yang diubah, Anda juga harus mengetahui pengaturan aslinya.

Saat Anda menyediakan infrastruktur CloudFormation, CloudFormation template menjelaskan dengan tepat sumber daya apa yang disediakan dan pengaturannya. Karena templat ini adalah file teks, Anda cukup melacak perbedaan dalam templat untuk melacak perubahan infrastruktur Anda, mirip dengan cara pengembang mengendalikan revisi ke kode sumber. Misalnya, Anda dapat menggunakan sistem kontrol versi dengan templat sehingga Anda tahu persis perubahan apa yang dibuat, siapa yang membuatnya, dan kapan dibuatnya. Jika suatu saat Anda perlu membalikkan perubahan pada infrastruktur, Anda dapat menggunakan versi templat sebelumnya.

Memulai dengan CloudFormation

CloudFormation tersedia melalui CloudFormation [konsol](#), [API](#), [AWS CLI](#) [AWS SDKs](#), dan melalui beberapa integrasi.

Untuk pengantar CloudFormation, lihat [Bagaimana cara CloudFormation kerja](#).

Untuk mulai menggunakan CloudFormation, lihat [Membuat tumpukan pertama Anda](#).

Informasi terkait

Anda dapat mempelajari lebih lanjut CloudFormation di panduan pengguna ini, serta sumber daya berikut:

- Untuk detail produk dan FAQs, lihat [halaman AWS CloudFormation produk](#).

- Untuk informasi harga, lihat [Harga AWS CloudFormation](#).

Memulai dengan CloudFormation

Anda dapat mulai menggunakan CloudFormation melalui AWS Management Console dengan membuat tumpukan dari contoh template, yang akan membantu Anda mempelajari dasar-dasar pembuatan tumpukan. Template adalah file teks yang mendefinisikan semua sumber daya dalam tumpukan. Tumpukan adalah penyebaran CloudFormation template. Dari satu template, Anda dapat membuat beberapa tumpukan. Setiap tumpukan berisi kumpulan AWS sumber daya yang dapat dikelola sebagai satu unit.

CloudFormation adalah layanan gratis; Namun, Anda dikenakan biaya untuk AWS sumber daya yang Anda sertakan dalam tumpukan Anda dengan tarif saat ini untuk masing-masing. Untuk informasi lebih lanjut tentang AWS harga, buka halaman detail untuk setiap produk di <http://aws.amazon.com>.

Topik

- [Bagaimana cara CloudFormation kerja](#)
- [Mendaftar untuk Akun AWS](#)
- [Membuat tumpukan pertama Anda](#)

Bagaimana cara CloudFormation kerja

Topik ini menjelaskan cara CloudFormation kerja dan memperkenalkan Anda pada konsep-konsep kunci yang perlu Anda ketahui saat Anda menggunakannya.

Topik

- [Konsep utama](#)
- [Bagaimana cara CloudFormation kerja](#)
- [Cara untuk memulai CloudFormation](#)

Konsep utama

Saat Anda menggunakan CloudFormation, Anda bekerja dengan templat dan tumpukan. Anda membuat template untuk menggambarkan AWS sumber daya Anda dan propertinya. Setiap kali Anda membuat tumpukan CloudFormation, berikan sumber daya yang dijelaskan dalam template Anda.

Topik

- [Template](#)
- [Tumpukan](#)
- [Ubah set](#)

Template

CloudFormation Template adalah file teks berformat YAMAL atau JSON. Anda dapat menyimpan file-file ini dengan ekstensi apa pun, seperti `.yaml`, `.json`, `.template`, atau `.txt`. CloudFormation menggunakan template ini sebagai cetak biru untuk membangun sumber daya Anda. AWS Misalnya, dalam template, Anda dapat mendeskripsikan EC2 instance Amazon, seperti jenis instans, ID AMI, pemetaan perangkat blok, dan nama EC2 key pair Amazon-nya. Setiap kali Anda membuat tumpukan, Anda juga menentukan templat yang CloudFormation gunakan untuk membuat apa pun yang Anda jelaskan dalam templat.

Misalnya, jika Anda membuat tumpukan dengan template berikut, berikan CloudFormation instance dengan ID `ami-0ff8a91507f77f867` AMI, jenis `t2.micro` instance, nama `testkey` key pair, dan volume Amazon EBS.

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
          }
        }
      ]
    }
  }
}

```

Anda juga dapat menentukan beberapa sumber daya dalam satu templat dan mengkonfigurasi sumber daya ini untuk bekerja sama. Misalnya, Anda dapat memodifikasi template sebelumnya untuk menyertakan alamat IP Elastis (EIP) dan mengaitkannya dengan EC2 instance Amazon, seperti yang ditunjukkan pada contoh berikut:

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"
        }
      }
    }
  }
}
```

```
}  
}
```

Template sebelumnya berpusat di sekitar satu EC2 instans Amazon; namun, CloudFormation template memiliki kemampuan tambahan yang dapat Anda gunakan untuk membangun kumpulan sumber daya yang kompleks dan menggunakan kembali template tersebut dalam berbagai konteks. Misalnya, Anda dapat menambahkan parameter input yang nilainya ditentukan saat Anda membuat CloudFormation tumpukan. Dengan kata lain, Anda dapat menentukan nilai seperti tipe instans ketika Anda membuat tumpukan bukan ketika Anda membuat templat, membuat templat lebih mudah untuk digunakan kembali dalam situasi yang berbeda.

Tumpukan

Saat Anda menggunakan CloudFormation, Anda mengelola sumber daya terkait sebagai satu unit yang disebut tumpukan. Anda membuat, memperbarui, dan menghapus kumpulan sumber daya dengan membuat, memperbarui, dan menghapus tumpukan. Semua sumber daya dalam tumpukan ditentukan oleh CloudFormation template stack. Misalkan Anda membuat template yang menyertakan grup Auto Scaling, penyeimbang beban Elastic Load Balancing, dan instance database Amazon Relational Database Service (Amazon RDS). Untuk membuat sumber daya tersebut, Anda membuat tumpukan dengan mengirimkan template yang Anda buat, dan CloudFormation menyediakan semua sumber daya tersebut untuk Anda.

Ubah set

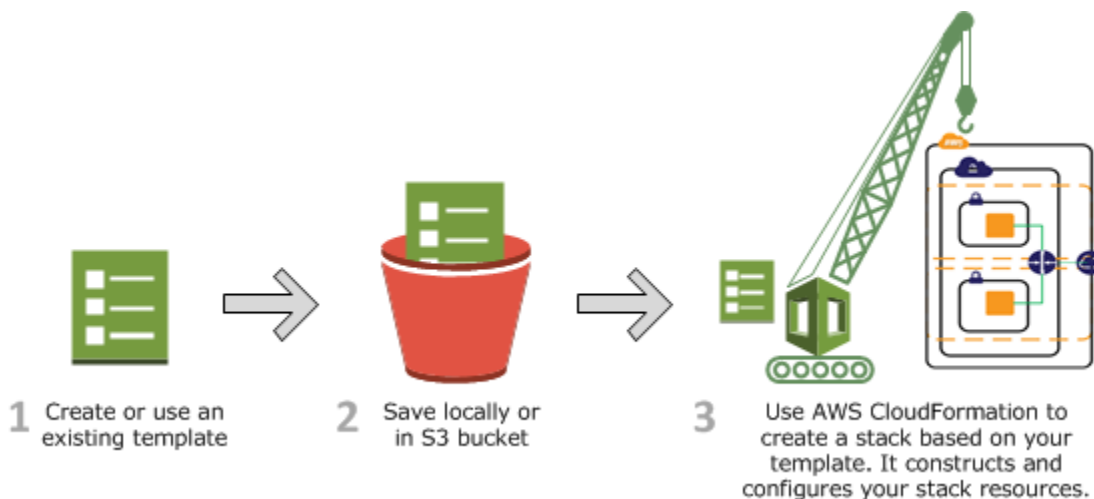
Jika Anda perlu mengubah sumber daya yang sedang berjalan dalam tumpukan, Anda memperbarui tumpukan. Sebelum membuat perubahan pada sumber daya Anda, Anda dapat membuat kumpulan perubahan, yang merupakan ringkasan perubahan yang Anda ajukan. Set perubahan memungkinkan Anda melihat bagaimana perubahan dapat mempengaruhi sumber daya yang sedang berjalan, terutama untuk sumber daya kritis, sebelum menerapkannya.

Misalnya, jika Anda mengubah nama instance database Amazon RDS, CloudFormation akan membuat database baru dan menghapus yang lama. Anda akan kehilangan data dalam basis data lama kecuali Anda sudah mencadangkannya. Jika Anda membuat set perubahan, Anda akan melihat bahwa perubahan Anda akan menyebabkan basis data Anda diganti, dan Anda akan dapat menyesuaikan rencana sebelum Anda memperbarui tumpukan Anda.

Bagaimana cara CloudFormation kerja

Saat Anda menggunakan CloudFormation untuk membuat tumpukan, lakukan CloudFormation panggilan layanan yang mendasarinya AWS ke penyedia dan mengonfigurasi sumber daya yang dijelaskan dalam templat Anda. Anda memerlukan izin untuk membuat sumber daya ini. Misalnya, untuk membuat EC2 instance dengan menggunakan CloudFormation, Anda memerlukan izin untuk membuat instance. Anda mengelola izin ini dengan [AWS Identity and Access Management](#) (IAM).

Panggilan yang dilakukan CloudFormation semuanya dideklarasikan oleh template Anda. Misalnya, Anda memiliki template yang menggambarkan sebuah EC2 instance dengan tipe `t2.micro` instance. Saat Anda menggunakan template tersebut untuk membuat tumpukan, CloudFormation memanggil Amazon EC2 create instance API dan menentukan jenis instance sebagai `t2.micro`. Diagram berikut merangkum alur kerja CloudFormation untuk membuat tumpukan.



Untuk membuat tumpukan

1. Gunakan editor teks untuk membuat CloudFormation template dalam format YAML atau JSON. Templat CloudFormation menjelaskan sumber daya yang Anda inginkan dan pengaturannya. Gunakan [Komposer Infrastruktur](#) untuk memvisualisasikan dan memvalidasi template Anda. Ini membantu Anda memastikan bahwa template Anda terstruktur dengan benar dan bebas dari kesalahan sintaks. Untuk informasi selengkapnya, lihat [Bekerja dengan CloudFormation template](#).
2. Simpan template secara lokal atau di bucket Amazon S3.
3. Buat CloudFormation tumpukan dengan menentukan lokasi file template Anda, seperti jalur di komputer lokal Anda atau URL Amazon S3. Jika templat berisi parameter, Anda dapat menentukan nilai input saat membuat tumpukan. Parameter memungkinkan Anda untuk

meneruskan nilai ke template Anda sehingga Anda dapat menyesuaikan sumber daya Anda setiap kali Anda membuat tumpukan.

Note

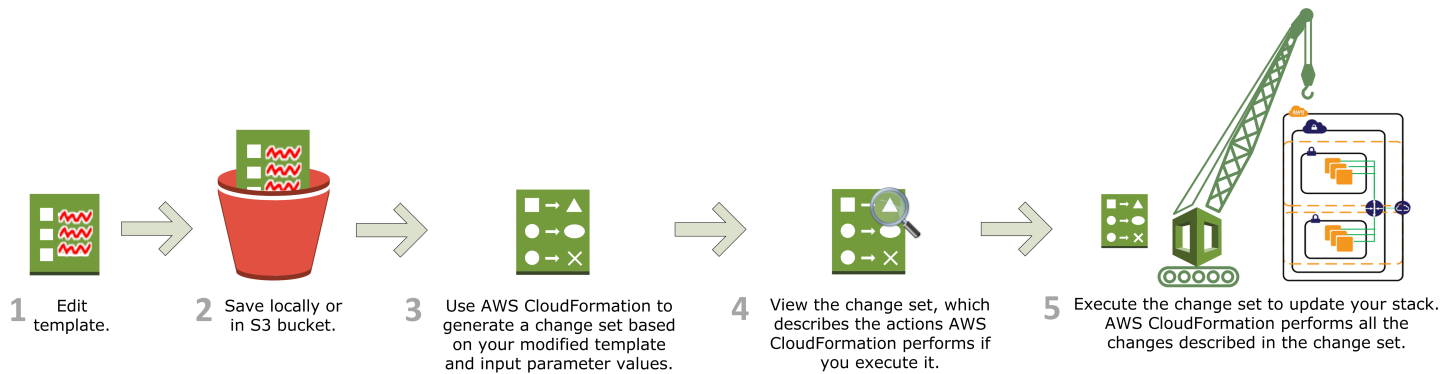
Jika Anda menentukan file templat yang disimpan secara lokal, CloudFormation unggah ke bucket S3 di akun Anda. AWS CloudFormation membuat bucket untuk setiap wilayah tempat Anda mengunggah file template. Bucket dapat diakses oleh siapa saja yang memiliki izin Amazon Simple Storage Service (Amazon S3) Simple Storage Service (Amazon S3) di akun Anda. AWS Jika bucket yang dibuat oleh CloudFormation sudah ada, template ditambahkan ke bucket itu.

Anda dapat menggunakan bucket Anda sendiri dan mengelola izinnya dengan mengunggah templat secara manual ke Amazon S3. Kemudian setiap kali Anda membuat atau memperbarui tumpukan, tentukan URL Amazon S3 dari file template.

Setelah semua sumber daya dibuat, CloudFormation laporkan bahwa tumpukan Anda telah dibuat. Kemudian, Anda dapat mulai menggunakan sumber daya dalam tumpukan Anda. Jika pembuatan tumpukan gagal, CloudFormation mengembalikan perubahan Anda dengan menghapus sumber daya yang dibuat.

Memperbarui tumpukan dengan set perubahan

Ketika Anda perlu memperbarui sumber daya tumpukan, Anda dapat mengubah templat tumpukan. Anda tidak perlu membuat tumpukan baru dan menghapus yang lama. Untuk memperbarui tumpukan, buat set perubahan dengan mengirimkan versi modifikasi dari template tumpukan asli, nilai parameter input yang berbeda, atau keduanya. CloudFormation membandingkan template yang dimodifikasi dengan template asli dan menghasilkan set perubahan. Set perubahan mencantumkan perubahan yang diajukan. Setelah meninjau perubahan, Anda dapat memulai set perubahan untuk memperbarui tumpukan Anda atau Anda dapat membuat set perubahan baru. Diagram berikut merangkum alur kerja untuk memperbarui tumpukan.



Untuk memperbarui tumpukan dengan set perubahan

1. Anda dapat memodifikasi template CloudFormation tumpukan dengan menggunakan [Komposer Infrastruktur](#) atau editor teks. Untuk informasi selengkapnya, lihat [Perbarui template tumpukan Anda](#).

Saat Anda memperbarui template Anda, perlu diingat bahwa pembaruan dapat menyebabkan gangguan. Bergantung pada sumber daya dan properti yang Anda perbarui, pembaruan mungkin mengganggu atau bahkan mengganti sumber daya yang ada. Untuk informasi selengkapnya, lihat [Memahami perilaku pembaruan sumber daya tumpukan](#).

2. Simpan CloudFormation template secara lokal atau dalam ember S3.
3. Buat set perubahan dengan menentukan tumpukan yang ingin Anda perbarui dan lokasi templat yang dimodifikasi, seperti jalur di komputer lokal Anda atau URL Amazon S3. Untuk informasi selengkapnya tentang cara membuat set perubahan, lihat [Perbarui CloudFormation tumpukan menggunakan set perubahan](#).

Note

Jika Anda menentukan template yang disimpan di komputer lokal Anda, CloudFormation secara otomatis mengunggah template Anda ke bucket S3 di komputer Anda. Akun AWS

4. Lihat set perubahan untuk memeriksa apakah CloudFormation akan melakukan perubahan yang Anda harapkan. Misalnya, periksa apakah CloudFormation akan mengganti sumber daya tumpukan penting. Anda dapat membuat set perubahan sebanyak yang dibutuhkan sampai Anda telah menyertakan perubahan yang diinginkan.

⚠ Important

Set perubahan tidak menunjukkan apakah pembaruan tumpukan Anda akan berhasil. Misalnya, set perubahan tidak memeriksa apakah Anda akan melampaui [kuota](#) akun, jika Anda memperbarui sumber daya yang tidak mendukung pembaruan, atau jika Anda memiliki [izin](#) yang tidak memadai untuk mengubah sumber daya, yang dapat menyebabkan pembaruan tumpukan gagal.

5. Mulai set perubahan yang ingin Anda terapkan ke tumpukan Anda. CloudFormation memperbarui tumpukan Anda dengan memperbarui hanya sumber daya yang Anda modifikasi dan memberi sinyal bahwa tumpukan Anda telah berhasil diperbarui. Jika pembaruan tumpukan gagal, CloudFormation memutar kembali perubahan untuk mengembalikan tumpukan ke status kerja terakhir yang diketahui.

Cara untuk memulai CloudFormation

Untuk membuat CloudFormation tumpukan hello world dengan konsol, lihat [Membuat tumpukan pertama Anda](#).

Untuk pembelajaran terpandu, cobalah [Memulai dengan AWS CloudFormation](#) lokakarya, yang menawarkan pengalaman langsung dengan pengembangan templat.

Mendaftar untuk Akun AWS

Ketika Anda mendaftar AWS, Anda Akun AWS secara otomatis mendaftar untuk semua layanan di AWS, termasuk AWS CloudFormation. Jika Anda Akun AWS sudah memiliki, lompat ke topik berikutnya. Jika Anda tidak memiliki Akun AWS, gunakan prosedur berikut untuk membuatnya.

Mendaftar untuk Akun AWS

Jika Anda tidak memiliki Akun AWS, selesaikan langkah-langkah berikut untuk membuatnya.

Untuk mendaftar untuk Akun AWS

1. Buka <https://portal.aws.amazon.com/billing/pendaftaran>.
2. Ikuti petunjuk online.

Bagian dari prosedur pendaftaran melibatkan menerima panggilan telepon atau pesan teks dan memasukkan kode verifikasi pada keypad telepon.

Saat Anda mendaftar untuk sebuah Akun AWS, sebuah Pengguna root akun AWS dibuat. Pengguna root memiliki akses ke semua Layanan AWS dan sumber daya di akun. Sebagai praktik keamanan terbaik, tetapkan akses administratif ke pengguna, dan gunakan hanya pengguna root untuk melakukan [tugas yang memerlukan akses pengguna root](#).

AWS mengirimkan Anda email konfirmasi setelah proses pendaftaran selesai. Kapan saja, Anda dapat melihat aktivitas akun Anda saat ini dan mengelola akun Anda dengan masuk <https://aws.amazon.com/ke/> dan memilih Akun Saya.

Buat pengguna dengan akses administratif

Setelah Anda mendaftar Akun AWS, amankan Pengguna root akun AWS, aktifkan AWS IAM Identity Center, dan buat pengguna administratif sehingga Anda tidak menggunakan pengguna root untuk tugas sehari-hari.

Amankan Anda Pengguna root akun AWS

1. Masuk ke [AWS Management Console](#) sebagai pemilik akun dengan memilih pengguna Root dan memasukkan alamat Akun AWS email Anda. Di laman berikutnya, masukkan kata sandi.

Untuk bantuan masuk dengan menggunakan pengguna root, lihat [Masuk sebagai pengguna root](#) di AWS Sign-In Panduan Pengguna.

2. Mengaktifkan autentikasi multi-faktor (MFA) untuk pengguna root Anda.

Untuk petunjuk, lihat [Mengaktifkan perangkat MFA virtual untuk pengguna Akun AWS root \(konsol\) Anda](#) di Panduan Pengguna IAM.

Buat pengguna dengan akses administratif

1. Aktifkan Pusat Identitas IAM.

Untuk mendapatkan petunjuk, silakan lihat [Mengaktifkan AWS IAM Identity Center](#) di Panduan Pengguna AWS IAM Identity Center .

2. Di Pusat Identitas IAM, berikan akses administratif ke pengguna.

Untuk tutorial tentang menggunakan Direktori Pusat Identitas IAM sebagai sumber identitas Anda, lihat [Mengkonfigurasi akses pengguna dengan default Direktori Pusat Identitas IAM](#) di Panduan AWS IAM Identity Center Pengguna.

Masuk sebagai pengguna dengan akses administratif

- Untuk masuk dengan pengguna Pusat Identitas IAM, gunakan URL masuk yang dikirim ke alamat email saat Anda membuat pengguna Pusat Identitas IAM.

Untuk bantuan masuk menggunakan pengguna Pusat Identitas IAM, lihat [Masuk ke portal AWS akses](#) di Panduan AWS Sign-In Pengguna.

Tetapkan akses ke pengguna tambahan

1. Di Pusat Identitas IAM, buat set izin yang mengikuti praktik terbaik menerapkan izin hak istimewa paling sedikit.

Untuk petunjuknya, lihat [Membuat set izin](#) di Panduan AWS IAM Identity Center Pengguna.

2. Tetapkan pengguna ke grup, lalu tetapkan akses masuk tunggal ke grup.

Untuk petunjuk, lihat [Menambahkan grup](#) di Panduan AWS IAM Identity Center Pengguna.

Note

Untuk informasi selengkapnya tentang cara mengelola siapa yang memiliki akses ke apa, lihat [Kontrol CloudFormation akses dengan AWS Identity and Access Management](#).

Membuat tumpukan pertama Anda

Topik ini memandu Anda melalui pembuatan CloudFormation tumpukan pertama Anda menggunakan AWS Management Console. Dengan mengikuti tutorial ini, Anda akan belajar cara menyediakan AWS sumber daya dasar, memantau peristiwa tumpukan, dan menghasilkan output.

Untuk contoh ini, CloudFormation template ditulis dalam YANG. YANG adalah format yang dapat dibaca manusia yang banyak digunakan untuk mendefinisikan infrastruktur sebagai kode. Saat Anda

mempelajari lebih lanjut CloudFormation, Anda mungkin juga menemukan template lain dalam format JSON, tetapi untuk tutorial ini, YANG dipilih karena keterbacaannya.

Note

CloudFormation gratis, tetapi Anda akan dikenakan biaya untuk sumber daya Amazon EC2 dan Amazon S3 yang Anda buat. Namun, jika Anda baru mengenal AWS, Anda dapat memanfaatkan [Tingkat Gratis](#) untuk meminimalkan atau menghilangkan biaya selama proses pembelajaran ini.

Topik

- [Prasyarat](#)
- [Buat CloudFormation tumpukan dengan konsol](#)
- [Pantau pembuatan tumpukan](#)
- [Uji server web](#)
- [Pemecahan Masalah](#)
- [Bersihkan](#)
- [Langkah selanjutnya](#)

Prasyarat

- Anda harus memiliki akses ke pengguna atau peran IAM yang memiliki izin untuk menggunakan Amazon EC2, Amazon S3, CloudFormation dan, atau akses pengguna administratif. Akun AWS
- Anda harus memiliki Virtual Private Cloud (VPC) yang memiliki akses ke internet. Template walkthrough ini membutuhkan VPC default, yang datang secara otomatis dengan yang lebih baru. Akun AWS Jika Anda tidak memiliki VPC default, atau jika telah dihapus, lihat bagian pemecahan masalah dalam topik ini untuk solusi alternatif.

Buat CloudFormation tumpukan dengan konsol

Untuk membuat CloudFormation tumpukan Hello world dengan konsol

1. Buka [konsol CloudFormation](#).
2. Pilih Buat tumpukan.

3. Pada halaman Create stack, pilih Build from Infrastructure Composer, dan kemudian Create in Infrastructure Composer. Ini membawa Anda ke Infrastructure Composer dalam mode CloudFormation konsol tempat Anda dapat mengunggah dan memvalidasi template contoh.
4. Untuk mengunggah dan memvalidasi template contoh, lakukan hal berikut:
 - a. Pilih Template. Kemudian, salin dan tempel CloudFormation templat berikut ke editor templat:

```
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

```

Sebelum Anda pindah ke langkah berikutnya, mari kita luangkan waktu sejenak untuk melihat template dan memahami beberapa CloudFormation konsep kunci.

- **Parameters** Bagian ini mendeklarasikan nilai yang dapat diteruskan ke template saat Anda membuat tumpukan. Sumber daya yang ditentukan kemudian dalam templat mereferensikan nilai-nilai ini dan menggunakan data. Parameter adalah cara yang efektif untuk menentukan informasi yang tidak ingin Anda simpan di template itu sendiri. Mereka juga merupakan cara untuk menentukan informasi yang mungkin unik untuk aplikasi atau konfigurasi tertentu yang Anda gunakan.
- Template mendefinisikan parameter berikut:
 - **LatestAmiId**— Mengambil ID AMI Amazon Linux 2 terbaru dari AWS Systems Manager Parameter Store.

- **InstanceType**— Memungkinkan pemilihan jenis EC2 instance (default:t2.micro, allowed:t3.micro,t2.micro).
- **MyIP**- Menentukan rentang alamat IP untuk akses HTTP (default: 0.0.0.0/0, memungkinkan akses dari IP apa pun).
- **Resources**Bagian ini berisi definisi AWS sumber daya yang ingin Anda buat dengan template. Deklarasi sumber daya adalah cara yang efisien untuk menentukan semua pengaturan konfigurasi ini sekaligus. Saat menyertakan deklarasi sumber daya dalam templat, Anda dapat membuat dan mengonfigurasi semua sumber daya yang dideklarasikan dengan menggunakan templat tersebut untuk membuat tumpukan. Anda juga dapat membuat tumpukan baru dari template yang sama untuk meluncurkan konfigurasi sumber daya yang identik.
- Template ini menciptakan sumber daya berikut:
 - **WebServerSecurityGroup**- Grup EC2 keamanan yang memungkinkan lalu lintas HTTP masuk pada port 80 dari rentang IP yang ditentukan.
 - **WebServer**— Sebuah EC2 contoh dengan konfigurasi berikut:
 - Menggunakan Amazon Linux 2 AMI terbaru
 - Menerapkan jenis contoh yang dipilih
 - Menambahkan WebServerSecurityGroup ke SecurityGroupIds properti
 - Termasuk skrip data pengguna untuk menginstal Apache HTTP Server
- Nama logis ditentukan di awal setiap sumber daya dan deklarasi parameter. Misalnya, WebServerSecurityGroup adalah nama logis yang ditetapkan ke sumber daya grup EC2 keamanan. RefFungsi ini kemudian digunakan untuk referensi sumber daya dan parameter dengan nama logisnya di bagian lain dari template. Ketika satu sumber daya mereferensikan sumber daya lain, ini menciptakan ketergantungan di antara mereka.
- **Outputs**Bagian ini mendefinisikan nilai kustom yang dikembalikan setelah pembuatan tumpukan. Anda dapat menggunakan nilai output untuk mengembalikan informasi dari sumber daya di tumpukan, seperti pengidentifikasi sumber daya atau URLs.
- Template mendefinisikan satu output:
 - **WebsiteURL**— URL server web yang dikerahkan, dibangun menggunakan nama DNS publik EC2 instance. JoinFungsi ini membantu menggabungkan tetap http:// dengan variabel PublicDnsName menjadi satu string, sehingga mudah untuk menampilkan URL lengkap server web.

b. Pilih Validasi untuk memastikan kode YAMM valid sebelum mengunggah template.

- c. Selanjutnya, pilih Buat template untuk membuat template dan menambahkannya ke ember S3.
 - d. Dari kotak dialog yang terbuka, buat catatan nama bucket S3 sehingga Anda dapat menghapusnya nanti. Kemudian, pilih Konfirmasi dan lanjutkan ke CloudFormation. Ini membawa Anda ke CloudFormation konsol tempat jalur S3 ke template Anda sekarang ditentukan.
5. Pada halaman Buat tumpukan, pilih Berikutnya.
 6. Pada halaman Tentukan detail tumpukan, ketikkan nama di bidang Nama tumpukan. Nama tumpukan tidak dapat berisi spasi. Untuk contoh ini, gunakan **MyTestStack**.
 7. Di bawah Parameter, tentukan nilai parameter sebagai berikut:
 - LatestAmild: Ini diatur secara default ke Amazon Linux 2 AMI terbaru.
 - InstanceType: Pilih salah satu t2.micro atau t3.micro untuk jenis instance. EC2

 Note

Jika Anda baru AWS, Anda dapat menggunakan tingkat gratis untuk meluncurkan dan menggunakan t2.micro instance secara gratis selama 12 bulan (di Wilayah yang tidak t2.micro tersedia, Anda dapat menggunakan t3.micro instance di bawah tingkat gratis).

- MyIP: Tentukan alamat IP publik Anda yang sebenarnya dengan akhiran. /32 /32Sufiks digunakan dalam notasi CIDR untuk menentukan bahwa satu alamat IP diperbolehkan. Ini pada dasarnya berarti memungkinkan lalu lintas ke dan dari alamat IP spesifik ini, dan tidak ada yang lain.
8. Pilih Berikutnya dua kali untuk pergi ke halaman Review dan create. Untuk tutorial ini, Anda dapat meninggalkan default pada halaman Configure stack options sebagaimana adanya.
 9. Tinjau informasi untuk tumpukan. Setelah Anda puas dengan pengaturan, pilih Kirim.

Pantau pembuatan tumpukan

Setelah Anda memilih Kirim, CloudFormation mulailah membuat sumber daya yang ditentukan dalam template. Tumpukan baru Anda **MyTestStack**, muncul dalam daftar di bagian atas CloudFormation konsol. Statusnya seharusnya CREATE_IN_PROGRESS. Anda dapat melihat status detail untuk tumpukan dengan melihat peristiwanya.

Untuk melihat peristiwa sebuah tumpukan

1. Di CloudFormation konsol, pilih tumpukan **MyTestStack** dalam daftar.
2. Pilih tab Peristiwa di panel detail klaster.

Konsol secara otomatis menyegarkan daftar peristiwa dengan peristiwa terbaru setiap 60 detik.

Tab Peristiwa menampilkan setiap langkah besar dalam pembuatan tumpukan yang diurutkan berdasarkan waktu setiap peristiwa dengan kejadian terbaru berada di atas.

Acara pertama (di bagian bawah daftar acara) adalah awal dari proses pembuatan tumpukan:

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

Berikutnya adalah peristiwa yang menandai awal dan penyelesaian penciptaan setiap sumber daya. Misalnya, pembuatan EC2 instance menghasilkan entri berikut:

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

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

CREATE_IN_PROGRESS Acara dicatat ketika CloudFormation melaporkan bahwa ia telah mulai membuat sumber daya. Peristiwa CREATE_COMPLETE dicatat ketika sumber daya berhasil dibuat.

Ketika CloudFormation telah berhasil membuat tumpukan, Anda akan melihat peristiwa berikut di bagian atas tab Events:

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

Jika tidak CloudFormation dapat membuat sumber daya, ia melaporkan CREATE_FAILED peristiwa dan, secara default, memutar kembali tumpukan dan menghapus sumber daya apa pun yang telah dibuat. Parameter kolom Alasan Status menampilkan masalah yang menyebabkan kegagalan.

Setelah tumpukan dibuat, Anda dapat pergi ke tab Resources untuk melihat EC2 instance dan grup keamanan yang Anda buat.

Uji server web

Setelah tumpukan berhasil dibuat, navigasikan ke tab Output di CloudFormation konsol. Cari bidang WebsiteUrl. Ini akan berisi URL publik EC2 instance Anda.

Buka browser dan buka URL yang tercantum di bawah WebsiteUrl. Anda akan melihat "Hello World!" pesan sederhana yang ditampilkan di browser.

Ini mengonfirmasi bahwa EC2 instans Anda menjalankan Apache HTTP Server dan menyajikan halaman web dasar.

Pemecahan Masalah

Jika Anda mengalami rollback selama pembuatan tumpukan, itu mungkin karena VPC yang hilang. Berikut cara mengatasi masalah ini.

Tidak ada VPC default yang tersedia

Template dalam panduan ini membutuhkan VPC default. Jika pembuatan tumpukan gagal karena kesalahan ketersediaan VPC atau subnet, Anda mungkin tidak memiliki VPC default di akun Anda. Anda memiliki opsi berikut:

- Buat VPC default baru — Anda dapat membuat VPC default baru melalui konsol VPC Amazon. Untuk petunjuknya, lihat [Membuat VPC default](#) di Panduan Pengguna Amazon VPC.
- Ubah template untuk menentukan subnet - Jika Anda memiliki VPC non-default, Anda dapat memodifikasi template untuk secara eksplisit menentukan VPC dan subnet. IDs Tambahkan parameter berikut ke template:

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

Kemudian, perbarui WebServer sumber daya untuk menyertakan ID subnet:

```
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
```

Saat membuat tumpukan, Anda harus menentukan subnet yang memiliki akses internet agar server web dapat dijangkau.

Bersihkan

Untuk memastikan Anda tidak dikenakan biaya layanan yang tidak diinginkan, Anda dapat membersihkannya dengan menghapus tumpukan dan sumber dayanya. Anda juga dapat menghapus bucket S3 yang menyimpan template stack.

Untuk menghapus tumpukan dan sumber dayanya

1. Buka [konsol CloudFormation](#).
2. Pada halaman Stacks, pilih opsi di sebelah nama tumpukan yang Anda buat (**MyTestStack**) dan kemudian pilih Hapus.
3. Saat diminta konfirmasi, pilih Hapus.
4. Pantau kemajuan proses penghapusan tumpukan pada tab Event. Status untuk **MyTestStack** perubahan ke DELETE_IN_PROGRESS. Ketika CloudFormation menyelesaikan penghapusan tumpukan, itu menghapus tumpukan dari daftar.

Jika Anda selesai bekerja dengan template contoh dan tidak lagi memerlukan bucket Amazon S3 Anda, hapus saja. Sebelum Anda dapat menghapus ember, Anda harus terlebih dahulu mengosongkannya. Mengosongkan ember menghapus semua objek di dalamnya.

Untuk mengosongkan dan menghapus bucket Amazon S3

1. Buka [konsol Amazon S3](#).
2. Di panel navigasi di sisi kiri konsol, pilih Bucket.
3. Dalam daftar Bucket, pilih opsi di sebelah nama bucket yang Anda buat untuk tutorial ini, lalu pilih Empty.
4. Di halaman Bucket kosong, konfirmasikan bahwa Anda ingin mengosongkan bucket dengan mengetik **permanently delete** ke dalam bidang teks, lalu pilih Kosongkan.

5. Pantau progres proses pengosongan bucket di halaman Bucket kosong: Status.
6. Untuk kembali ke daftar bucket Anda, pilih Keluar.
7. Pilih opsi di sebelah nama bucket, lalu pilih Hapus.
8. Saat diminta konfirmasi, ketikkan nama bucket lalu pilih Delete bucket.
9. Pantau perkembangan proses penghapusan bucket dari daftar Bucket. Saat Amazon S3 menyelesaikan penghapusan bucket, itu akan menghapus bucket dari daftar.

Langkah selanjutnya

Selamat! Anda berhasil membuat tumpukan, memantau ciptaannya, dan menggunakan outputnya.

Untuk terus belajar:

- Pelajari lebih lanjut tentang templat sehingga Anda dapat membuatnya sendiri. Untuk informasi selengkapnya, lihat [Bekerja dengan CloudFormation template](#).
- Coba AWS CloudFormation lokakarya [Memulai dengan](#) lebih banyak praktik langsung dengan pembuatan template.
- Untuk versi singkat dari [Memulai dengan AWS CloudFormation](#), lihat [Menyebarkan aplikasi di Amazon EC2](#). Topik ini menjelaskan skenario yang sama menggunakan skrip CloudFormation pembantu, `cfn-init`, untuk mem-bootstrap EC2 instance Amazon.

CloudFormation praktik terbaik

Praktik terbaik adalah rekomendasi yang dapat membantu Anda menggunakan CloudFormation lebih efektif dan aman di seluruh alur kerjanya. Pelajari cara merencanakan dan mengatur tumpukan Anda, membuat templat yang menggambarkan sumber daya dan aplikasi perangkat lunak yang berjalan di dalamnya, serta mengelola tumpukan dan sumber dayanya. Praktik terbaik berikut didasarkan pada pengalaman dunia nyata dari CloudFormation pelanggan saat ini.

Perencanaan dan pengorganisasian

- [Mempersingkat loop umpan balik untuk meningkatkan kecepatan pengiriman](#)
- [Atur tumpukan Anda berdasarkan siklus hidup dan kepemilikan](#)
- [Gunakan referensi lintas tumpukan untuk mengeksport sumber daya bersama](#)
- [Gunakan IAM untuk mengontrol akses](#)
- [Gunakan kembali templat untuk mereplikasi tumpukan di beberapa lingkungan](#)
- [Verifikasi kuota untuk semua jenis sumber daya](#)
- [Gunakan modul untuk menggunakan kembali konfigurasi sumber daya](#)

Membuat templat

- [Jangan menanamkan kredensial dalam templat Anda](#)
- [Gunakan AWS tipe parameter -spesifik](#)
- [Gunakan batasan parameter](#)
- [Gunakan parameter semu untuk mempromosikan portabilitas](#)
- [Gunakan `AWS::CloudFormation::Init` untuk menyebarkan aplikasi perangkat lunak di instans Amazon EC2](#)
- [Gunakan skrip pembantu terbaru](#)
- [Validasi templat sebelum digunakan](#)

Mengelola tumpukan

- [Kelola semua sumber daya tumpukan melalui CloudFormation](#)
- [Buat rangkaian perubahan sebelum memperbarui tumpukan](#)
- [Gunakan kebijakan tumpukan](#)
- [Gunakan AWS CloudTrail untuk mencatat CloudFormation panggilan](#)
- [Gunakan ulasan kode dan kontrol revisi untuk mengelola templat](#)
- [Perbarui EC2 instans Amazon Anda secara teratur](#)

Mempersingkat loop umpan balik untuk meningkatkan kecepatan pengiriman

Mengadopsi praktik dan alat yang membantu Anda mempersingkat loop umpan balik untuk infrastruktur yang Anda gambarkan dengan CloudFormation template. Ini termasuk melakukan linting awal dan pengujian template Anda di workstation Anda; ketika Anda melakukannya, Anda memiliki kesempatan untuk menemukan potensi masalah sintaks dan konfigurasi bahkan sebelum Anda mengirimkan kontribusi Anda ke repositori kode sumber. Penemuan awal masalah tersebut membantu mencegah mereka mencapai lingkungan siklus hidup formal, seperti pengembangan, jaminan kualitas, dan produksi. Pendekatan pengujian awal dan cepat gagal ini memberi Anda manfaat mengurangi waktu tunggu pengerjaan ulang, mengurangi area dampak potensial, dan meningkatkan tingkat kepercayaan Anda dalam menjalankan operasi penyediaan yang berhasil.

Pilihan perkakas yang membantu Anda mencapai praktik cepat gagal termasuk [AWS CloudFormation Linter](#) (`cfn-lint`) dan [TaskCatalat](#) baris perintah. `cfn-lint` Alat ini memberi Anda kemampuan untuk memvalidasi CloudFormation template Anda terhadap [Spesifikasi AWS CloudFormation Sumber Daya](#). Ini termasuk memeriksa nilai yang valid untuk properti sumber daya, serta praktik terbaik. Plugin untuk `cfn-lint` [tersedia untuk sejumlah editor kode](#); ini memberi Anda kemampuan untuk memvisualisasikan masalah dalam editor Anda dan untuk mendapatkan umpan balik linter langsung. Anda juga dapat memilih untuk mengintegrasikan `cfn-lint` dalam konfigurasi repositori kode sumber Anda, sehingga Anda dapat melakukan validasi template ketika Anda melakukan kontribusi Anda. Untuk informasi selengkapnya, lihat [validasi pra-komit Git dari AWS CloudFormation template dengan cfn-lint](#). Setelah Anda melakukan linting awal Anda—dan memperbaiki masalah apa pun yang `cfn-lint` mungkin ditimbulkan—Anda dapat menggunakannya TaskCat untuk menguji template Anda dengan membuat tumpukan secara terprogram dalam pilihan Anda. Wilayah AWS TaskCat juga menghasilkan laporan dengan pass/fail nilai untuk setiap Wilayah yang Anda pilih.

[Untuk panduan langsung tentang cara menggunakan kedua alat untuk mempersingkat loop umpan balik, ikuti lab Linting dan Pengujian Lokakarya. step-by-step AWS CloudFormation](#)

Atur tumpukan Anda berdasarkan siklus hidup dan kepemilikan

Gunakan siklus hidup dan kepemilikan sumber daya Anda untuk membantu Anda memutuskan AWS sumber daya apa yang harus masuk di setiap tumpukan. Awalnya, Anda mungkin menempatkan semua sumber daya dalam satu tumpukan, tetapi seiring dengan bertambahnya skala dan cakupannya, mengelola satu tumpukan menjadi rumit dan memakan waktu. Dengan mengelompokkan sumber daya dengan siklus hidup umum dan kepemilikan, pemilik dapat membuat

perubahan pada kumpulan sumber daya dengan menggunakan proses dan jadwalnya sendiri tanpa mempengaruhi sumber daya lainnya.

Misalnya, bayangkan tim pengembang dan insinyur yang memiliki situs web yang di-host di instans EC2 Auto Scaling Amazon di belakang penyeimbang beban. Karena situs web memiliki siklus hidup sendiri dan dikelola oleh tim situs web, Anda dapat membuat tumpukan untuk situs web dan sumber dayanya. Sekarang bayangkan bahwa situs web juga menggunakan basis data back-end, di mana basis data berada dalam tumpukan terpisah yang dimiliki dan dikelola oleh administrator database. Setiap kali tim situs web atau tim basis data harus memperbarui sumber dayanya, hal tersebut dapat dilakukan tanpa mempengaruhi tumpukan masing-masing. Jika semua sumber daya berada di tumpukan tunggal, mengoordinasikan, dan mengomunikasikan pembaruan bisa jadi sulit.

Untuk panduan tambahan tentang mengatur tumpukan, Anda dapat menggunakan dua kerangka kerja umum: arsitektur berlapis-lapis dan arsitektur berorientasi layanan (SOA).

Arsitektur berlapis mengatur tumpukan menjadi beberapa lapisan horizontal yang dibangun di atas satu sama lain, di mana setiap lapisan memiliki ketergantungan pada lapisan tepat di bawahnya. Anda dapat memiliki satu tumpukan atau lebih di setiap lapisan, tetapi dalam setiap lapisan, tumpukan Anda harus memiliki AWS sumber daya dengan siklus hidup dan kepemilikan yang sama.

Dengan arsitektur berorientasi layanan, Anda dapat mengatur masalah bisnis besar menjadi bagian-bagian yang mudah dikelola. Masing-masing bagian ini adalah layanan yang memiliki tujuan yang jelas dan mewakili unit fungsionalitas yang berdiri mandiri. Anda dapat memetakan layanan ini ke tumpukan yang memiliki siklus hidup dan pemiliknya masing-masing. Layanan ini (tumpukan) dapat disatukan agar dapat berinteraksi satu sama lain.

Gunakan referensi lintas tumpukan untuk mengekspor sumber daya bersama

Saat Anda mengatur AWS sumber daya berdasarkan siklus hidup dan kepemilikan, Anda mungkin ingin membuat tumpukan yang menggunakan sumber daya yang ada di tumpukan lain. Anda dapat hardcode nilai atau menggunakan parameter input untuk meneruskan nama sumber daya dan IDs. Namun, metode ini dapat membuat templat sulit untuk digunakan kembali atau dapat meningkatkan overhead agar tumpukan bisa berjalan. Sebaliknya, gunakan referensi lintas tumpukan untuk mengekspor sumber daya dari tumpukan sehingga tumpukan lain dapat menggunakannya. Tumpukan dapat menggunakan sumber daya yang diekspor dengan memanggilnya menggunakan `Fn::ImportValue` fungsi.

Misalnya, Anda mungkin memiliki tumpukan jaringan yang mencakup VPC, grup keamanan, dan subnet. Anda ingin semua aplikasi web publik menggunakan sumber daya ini. Dengan mengekspor sumber daya, Anda mengizinkan semua tumpukan dengan aplikasi web publik menggunakannya. Untuk informasi selengkapnya, lihat [Dapatkan output yang diekspor dari tumpukan yang diterapkan CloudFormation](#).

Verifikasi kuota untuk semua jenis sumber daya

Sebelum meluncurkan tumpukan, pastikan Anda dapat membuat semua sumber daya yang Anda inginkan tanpa menekan batas AWS akun Anda. Jika Anda mencapai batas, tidak CloudFormation akan berhasil membuat tumpukan hingga Anda meningkatkan kuota atau menghapus sumber daya tambahan. Setiap layanan dapat memiliki berbagai batasan yang harus Anda ketahui sebelum meluncurkan tumpukan. Misalnya, secara default, Anda hanya dapat meluncurkan 2000 CloudFormation tumpukan per Wilayah di wilayah Anda Akun AWS. Untuk informasi selengkapnya tentang batas dan cara meningkatkan batas default, lihat [kuota AWS layanan](#) di Referensi Umum AWS.

Gunakan kembali templat untuk mereplikasi tumpukan di beberapa lingkungan

Setelah mengatur tumpukan dan sumber daya, Anda dapat menggunakan kembali templat untuk mereplikasi infrastruktur di beberapa lingkungan. Misalnya, Anda dapat membuat lingkungan untuk pengembangan, pengujian, dan produksi sehingga Anda dapat menguji perubahan sebelum menerapkannya ke dalam produksi. Agar templat dapat digunakan kembali, gunakan parameter, pemetaan, dan kondisi bagian sehingga Anda dapat menyesuaikan tumpukan ketika membuatnya. Misalnya, untuk lingkungan pengembangan, Anda dapat menentukan tipe instans biaya yang lebih rendah yang dibandingkan dengan lingkungan produksi, tetapi semua konfigurasi dan pengaturan lainnya tetap sama. Untuk informasi selengkapnya tentang parameter, pemetaan, dan kondisi, lihat [CloudFormation bagian template](#).

Gunakan modul untuk menggunakan kembali konfigurasi sumber daya

Seiring pertumbuhan infrastruktur, pola umum dapat muncul saat Anda mendeklarasikan komponen yang sama di setiap templat. Modul adalah cara Anda untuk mengemas konfigurasi sumber daya

untuk dimasukkan di seluruh tumpukan templat, dengan cara yang transparan, mudah dikelola, dan berulang. Modul dapat merangkum konfigurasi layanan umum dan praktik terbaik sebagai blok bangunan modular yang dapat disesuaikan untuk Anda masukkan dalam templat tumpukan Anda.

Blok bangunan ini dapat digunakan untuk satu sumber daya, seperti praktik terbaik untuk mendefinisikan instance Amazon Elastic Compute Cloud EC2 (Amazon), atau dapat digunakan untuk beberapa sumber daya, untuk menentukan pola umum arsitektur aplikasi. Blok bangunan ini dapat disarangkan ke modul lain, sehingga Anda dapat menumpuk praktik terbaik Anda ke dalam blok bangunan tingkat yang lebih tinggi. CloudFormation modul tersedia di [CloudFormation registri](#), sehingga Anda dapat menggunakannya seperti sumber daya asli. Saat Anda menggunakan CloudFormation modul, template modul diperluas ke template konsumsi, yang memungkinkan Anda mengakses sumber daya di dalam modul menggunakan [Ref](#) atau [Fn::GetAtt](#). Untuk informasi selengkapnya, lihat [Buat konfigurasi sumber daya yang dapat digunakan kembali yang dapat disertakan di seluruh templat dengan modul CloudFormation](#).

Gunakan AWS tipe parameter -spesifik

Jika template Anda memerlukan input untuk nilai AWS-spesifik yang ada, seperti Amazon Virtual Private Cloud yang ada IDs atau nama EC2 key pair Amazon, gunakan tipe parameter AWS-specific. Misalnya, Anda dapat menentukan parameter sebagai tipe `AWS::EC2::KeyPair::KeyName`, yang mengambil nama key pair yang ada di dalam Akun AWS dan di Wilayah tempat Anda membuat tumpukan. CloudFormation dapat dengan cepat memvalidasi nilai untuk tipe parameter AWS-spesifik sebelum membuat tumpukan Anda. Selain itu, jika Anda menggunakan CloudFormation konsol, CloudFormation menampilkan daftar drop-down nilai yang valid, sehingga Anda tidak perlu mencari atau menghafal nama VPC atau key IDs pair yang benar. Untuk informasi selengkapnya, lihat [Tentukan sumber daya yang ada saat runtime dengan CloudFormation tipe parameter yang disediakan](#).

Gunakan batasan parameter

Dengan kendala, Anda dapat menjelaskan nilai input yang diizinkan sehingga CloudFormation menangkap nilai yang tidak valid sebelum membuat tumpukan. Anda dapat mengatur kendala seperti panjang minimum, panjang maksimum, dan pola yang diperbolehkan. Misalnya, Anda dapat mengatur kendala pada nilai nama pengguna basis data sehingga panjang minimal harus delapan karakter dan hanya berisi karakter alfanumerik. Untuk informasi selengkapnya, lihat [CloudFormation Parameters sintaks template](#).

Gunakan parameter semu untuk mempromosikan portabilitas

Anda dapat menggunakan [parameter semu](#) dalam template Anda sebagai argumen untuk [fungsi intrinsik](#), seperti `Ref Fn::Sub`. Parameter pseudo adalah parameter yang telah ditentukan sebelumnya oleh CloudFormation Anda tidak menyatakannya di templat Anda. Menggunakan parameter semu dalam fungsi intrinsik meningkatkan portabilitas template tumpukan Anda di seluruh Wilayah dan akun.

Misalnya, bayangkan Anda ingin membuat templat di mana, untuk properti sumber daya tertentu, Anda perlu menentukan [Nama Sumber Daya Amazon](#) (ARN) dari sumber daya lain yang ada. Dalam hal ini, sumber daya yang ada adalah sumber daya [AWS Systems Manager Parameter Store](#) dengan ARN berikut: `arn:aws:ssm:us-east-1:123456789012:parameter/MySampleParameter`. Anda perlu menyesuaikan [format ARN](#) ke AWS partisi target, Wilayah, dan ID akun Anda. Alih-alih hard-coding nilai-nilai ini, Anda dapat menggunakan `AWS::Partition`, `AWS::Region`, dan parameter `AWS::AccountId` semu untuk membuat template Anda lebih portabel. Dalam hal ini, contoh berikut menunjukkan cara menggabungkan elemen dalam ARN dengan: CloudFormation !
Sub 'arn:\${AWS::Partition}:ssm:\${AWS::Region}:\${AWS::AccountId}:parameter/MySampleParameter'

Untuk contoh lain, asumsikan Anda ingin berbagi sumber daya atau konfigurasi di beberapa tumpukan. Dalam contoh ini, asumsikan Anda telah membuat [subnet](#) untuk VPC Anda, dan kemudian mengekspor ID-nya untuk digunakan dengan tumpukan lain di wilayah yang sama dan Akun AWS. Di tumpukan lain, Anda mereferensikan nilai yang diekspor dari subnet ID saat mendeskripsikan instance Amazon. EC2 Untuk contoh rinci menggunakan bidang `Export` output dan fungsi `Fn::ImportValue` intrinsik, lihat. [Lihat output sumber daya di tumpukan lain CloudFormation](#)

Ekspor tumpukan harus unik per akun dan Wilayah. Jadi, dalam hal ini, Anda dapat menggunakan parameter `AWS::StackName` semu untuk membuat awalan untuk ekspor Anda. Karena nama tumpukan juga harus unik per akun dan Wilayah, penggunaan parameter semu ini sebagai awalan meningkatkan kemungkinan memiliki nama ekspor yang unik sambil juga mempromosikan pendekatan yang dapat digunakan kembali di seluruh tumpukan tempat Anda mengekspor nilai. Atau, Anda dapat menggunakan awalan pilihan Anda sendiri.

Gunakan **AWS::CloudFormation::Init** untuk menyebarkan aplikasi perangkat lunak di instans Amazon EC2

Saat meluncurkan tumpukan, Anda dapat menginstal dan mengonfigurasi aplikasi perangkat lunak di EC2 instans Amazon dengan menggunakan skrip `cfn-init` pembantu dan sumber daya.

AWS::CloudFormation::Init Dengan menggunakan **AWS::CloudFormation::Init**, Anda dapat menjelaskan konfigurasi yang Anda inginkan daripada menuliskan langkah-langkah prosedural. Anda juga dapat memperbarui konfigurasi tanpa menciptakan ulang instans. Dan jika ada yang salah dengan konfigurasi Anda, CloudFormation buat log yang dapat Anda gunakan untuk menyelidiki masalah.

Dalam templat Anda, tentukan keadaan instalasi dan konfigurasi di **AWS::CloudFormation::Init** sumber daya. Untuk panduan yang menunjukkan cara menggunakan `cfn-init` dan **AWS::CloudFormation::Init**, lihat [Menyebarkan aplikasi di Amazon EC2](#)

Gunakan skrip pembantu terbaru

Skrip CloudFormation pembantu diperbarui secara berkala. Pastikan Anda menyertakan perintah berikut di `UserData` properti templat Anda sebelum memanggil skrip pembantu untuk memastikan bahwa instans yang Anda luncurkan mendapatkanskrip pembantu terbaru:

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

Untuk informasi selengkapnya tentang mendapatkan skrip pembantu terbaru, lihat [referensi skrip CloudFormation pembantu di Panduan Referensi Template](#).AWS CloudFormation

Validasi templat sebelum digunakan

Sebelum Anda menggunakan template untuk membuat atau memperbarui tumpukan, Anda dapat menggunakannya CloudFormation untuk memvalidasinya. Memvalidasi template dapat membantu Anda menangkap sintaks dan beberapa kesalahan semantik, seperti dependensi melingkar, sebelum membuat sumber daya apa pun. CloudFormation Jika Anda menggunakan CloudFormation konsol, konsol secara otomatis memvalidasi template setelah Anda menentukan parameter input. Untuk CloudFormation API AWS CLI atau, gunakan perintah [validate-template](#)CLI atau operasi [ValidateTemplate](#)API.

Selama validasi, periksa CloudFormation terlebih dahulu apakah template adalah JSON yang valid. Jika bukan, CloudFormation memeriksa apakah templat YAML valid. Jika kedua pemeriksaan gagal, CloudFormation mengembalikan kesalahan validasi template.

Validasi template untuk kepatuhan kebijakan organisasi

Anda juga dapat memvalidasi template Anda untuk kepatuhan terhadap pedoman kebijakan organisasi. AWS CloudFormation Guard (`cfn-guard`) adalah alat antarmuka baris perintah sumber terbuka (CLI) yang menyediakan bahasa untuk menentukan aturan policy-as-code yang dapat memeriksa konfigurasi sumber daya yang diperlukan dan yang dilarang. Hal ini kemudian memungkinkan Anda untuk memvalidasi templat terhadap aturan-aturan tersebut. Misalnya, administrator dapat membuat aturan untuk memastikan bahwa pengguna selalu membuat bucket S3 Amazon terenkripsi.

Anda dapat menggunakan `cfn-guard` secara lokal, saat mengedit templat, atau secara otomatis sebagai bagian dari CI/CD pipeline untuk menghentikan penyebaran sumber daya yang tidak sesuai.

Selain itu, `cfn-guard` termasuk fitur `rulegen`, yang memungkinkan Anda untuk mengekstrak aturan dari CloudFormation template yang sesuai yang ada.

Untuk informasi lebih lanjut, lihat repositori [cfn-guard](#) di GitHub.

Kelola semua sumber daya tumpukan melalui CloudFormation

Setelah Anda meluncurkan tumpukan, gunakan CloudFormation [konsol](#), [API](#), atau [AWS CLI](#) untuk memperbarui sumber daya di tumpukan Anda. Jangan membuat perubahan pada tumpukan sumber daya di luar CloudFormation. Melakukannya dapat membuat ketidakcocokan antara template tumpukan Anda dan keadaan sumber daya tumpukan Anda saat ini, yang dapat menyebabkan kesalahan jika Anda memperbarui atau menghapus tumpukan. Ini dikenal sebagai drift. Jika perubahan dibuat ke sumber daya di luar CloudFormation template dan Anda memperbarui tumpukan, perubahan yang dibuat langsung ke sumber daya akan dibuang, dan konfigurasi sumber daya akan kembali ke konfigurasi di template.

Untuk informasi lebih lanjut tentang drift, lihat [Apa yang dimaksud dengan penyimpangan?](#).

Untuk informasi selengkapnya tentang memperbarui tumpukan, lihat [Memperbarui tumpukan](#).

Buat rangkaian perubahan sebelum memperbarui tumpukan

Set perubahan memungkinkan Anda melihat bagaimana perubahan yang diusulkan pada tumpukan dapat memengaruhi sumber daya yang sedang berjalan sebelum Anda menerapkannya. CloudFormation tidak membuat perubahan apa pun pada tumpukan Anda sampai Anda menjalankan set perubahan, memungkinkan Anda memutuskan apakah akan melanjutkan perubahan yang Anda usulkan atau membuat set perubahan lain.

Gunakan set perubahan untuk memeriksa bagaimana perubahan dapat mempengaruhi sumber daya yang sedang berjalan, terutama untuk sumber daya kritis. Misalnya, jika Anda mengubah nama instance database Amazon RDS, CloudFormation akan membuat database baru dan menghapus yang lama; Anda akan kehilangan data di database lama kecuali Anda sudah mencadangkannya. Jika Anda menghasilkan perubahan set, Anda akan melihat bahwa perubahan akan menggantikan basis data. Hal ini dapat membantu merencanakan sebelum memperbarui tumpukan. Untuk informasi selengkapnya, lihat [Perbarui CloudFormation tumpukan menggunakan set perubahan](#).

Gunakan kebijakan tumpukan

Kebijakan tumpukan membantu melindungi sumber daya tumpukan penting dari pembaruan yang tidak disengaja yang dapat menyebabkan sumber daya terganggu atau bahkan diganti. Kebijakan tumpukan adalah dokumen JSON yang menjelaskan tindakan pembaruan apa yang dapat dilakukan pada sumber daya yang ditunjuk. Tentukan kebijakan tumpukan setiap kali Anda membuat tumpukan yang memiliki sumber daya penting.

Selama pembaruan tumpukan, Anda harus secara eksplisit menentukan sumber daya yang dilindungi yang ingin Anda perbarui; jika tidak, tidak ada perubahan yang dibuat untuk sumber daya yang dilindungi. Untuk informasi selengkapnya, lihat [Mencegah pembaruan ke sumber daya tumpukan](#).

Gunakan ulasan kode dan kontrol revisi untuk mengelola templat

Templat tumpukan Anda menjelaskan konfigurasi sumber AWS daya Anda, seperti nilai propertinya. Untuk meninjau perubahan dan menyimpan riwayat sumber daya yang tepat, gunakan ulasan kode dan kontrol revisi. Metode ini dapat membantu Anda melacak perubahan antara versi templat yang berbeda, yang dapat membantu Anda melacak perubahan pada sumber daya tumpukan Anda. Selain itu, dengan mempertahankan riwayat, Anda dapat selalu mengembalikan tumpukan Anda ke versi khusus templat Anda.

Perbarui EC2 instans Amazon Anda secara teratur

Pada semua instans Amazon EC2 Windows dan instans Amazon EC2 Linux yang dibuat dengan CloudFormation, jalankan `yum update` perintah secara teratur untuk memperbarui paket RPM. Hal ini memastikan bahwa Anda mendapatkan perbaikan terbaru dan pembaruan keamanan.

Bekerja dengan CloudFormation template

AWS CloudFormation Template mendefinisikan AWS sumber daya yang ingin Anda buat, perbarui, atau hapus sebagai bagian dari tumpukan. Ini terdiri dari beberapa bagian, tetapi satu-satunya bagian yang diperlukan adalah [Resources](#) bagian, yang harus menyatakan setidaknya satu sumber daya.

Anda dapat membuat template menggunakan metode berikut:

- AWS Infrastructure Composer- Antarmuka visual untuk merancang template.
- Editor Teks - Tulis template langsung dalam sintaks JSON atau YAML.
- Generator IAC — Hasilkan templat dari sumber daya yang disediakan di akun Anda yang saat ini tidak dikelola oleh. CloudFormation Generator IAC bekerja dengan berbagai jenis sumber daya yang didukung oleh Cloud Control API di Wilayah Anda.

Bagian ini memberikan panduan komprehensif tentang cara menggunakan bagian yang berbeda dari CloudFormation template dan cara mulai membuat template tumpukan. Ini mencakup topik-topik berikut:

Topik

- [Tempat template disimpan](#)
- [Memvalidasi template](#)
- [Memulai dengan template](#)
- [Sampel template](#)
- [CloudFormation format templat](#)
- [CloudFormation bagian template](#)
- [Buat template secara visual dengan Infrastructure Composer](#)
- [Hasilkan templat dari sumber daya yang ada dengan generator IAC](#)
- [Dapatkan nilai yang disimpan di layanan lain menggunakan referensi dinamis](#)
- [Dapatkan AWS nilai menggunakan parameter semu](#)
- [Dapatkan output yang diekspor dari tumpukan yang diterapkan CloudFormation](#)
- [Tentukan sumber daya yang ada saat runtime dengan CloudFormation tipe parameter yang disediakan](#)
- [CloudFormation penelusuran](#)

- [CloudFormation cuplikan template](#)
- [Menyebarkan tumpukan Windows berbasis menggunakan CloudFormation](#)
- [Perluas kemampuan template Anda dengan CloudFormation tipe sumber daya yang disediakan](#)
- [Buat konfigurasi sumber daya yang dapat digunakan kembali yang dapat disertakan di seluruh templat dengan modul CloudFormation](#)

Tempat template disimpan

Bucket Amazon S3

Anda dapat menyimpan CloudFormation template di ember Amazon S3. Saat membuat atau memperbarui tumpukan, Anda dapat menentukan URL S3 template alih-alih mengunggahnya secara langsung.

Jika Anda mengunggah template langsung melalui AWS Management Console or AWS CLI, bucket S3 secara otomatis dibuat untuk Anda. Untuk informasi selengkapnya, lihat [Buat tumpukan dari CloudFormation konsol](#).

Repositori Git

Dengan [sinkronisasi Git](#), Anda dapat menyimpan template dalam repositori Git. Saat membuat atau memperbarui tumpukan, Anda dapat menentukan lokasi repositori Git dan cabang yang berisi templat alih-alih mengunggahnya secara langsung atau mereferensikan URL S3. CloudFormation secara otomatis memonitor repositori dan cabang yang ditentukan untuk perubahan template. Untuk informasi selengkapnya, lihat [Buat tumpukan dari kode sumber repositori dengan sinkronisasi Git](#).

Memvalidasi template

Validasi sintaks

Anda dapat memverifikasi sintaks JSON atau YAMAL template Anda dengan menggunakan perintah [validate-template](#) CLI atau dengan menentukan template Anda di konsol. Konsol melakukan validasi secara otomatis. Untuk informasi selengkapnya, lihat [Buat tumpukan dari CloudFormation konsol](#).

Namun, metode ini hanya memverifikasi sintaks template Anda dan tidak memvalidasi nilai properti yang Anda tentukan untuk sumber daya.

Alat validasi tambahan

Untuk validasi yang lebih kompleks dan pemeriksaan praktik terbaik, Anda dapat menggunakan alat tambahan seperti:

- [CloudFormation Linter \(cfn-lint\)](#) — Validasi template terhadap skema penyedia sumber daya. [CloudFormation](#) Termasuk memeriksa nilai yang valid untuk properti sumber daya dan praktik terbaik.
- [CloudFormation Rain \(rain fmt\)](#) - Format CloudFormation template Anda ke standar yang konsisten atau format ulang template dari JSON ke YAMAL (atau YAMAL ke JSON). Ini mempertahankan komentar saat menggunakan YAMAL dan mengalihkan penggunaan fungsi intrinsik ke sintaks pendek jika memungkinkan.

Memulai dengan template

Untuk memulai membuat CloudFormation template, ikuti langkah-langkah berikut:

1. Pilih sumber daya — Identifikasi AWS sumber daya yang ingin Anda sertakan dalam tumpukan Anda, seperti EC2 instans VPCs, grup keamanan, dan lainnya.
2. Tulis template - Tulis template dalam format JSON atau YAMAL, tentukan sumber daya dan propertinya.
3. Simpan template - Simpan template secara lokal dengan ekstensi file seperti: `.json`, `.yaml`, atau `.txt`.
4. Validasi template - Validasi template menggunakan metode yang dijelaskan di bagian. [Memvalidasi template](#)
5. Buat tumpukan — Buat tumpukan menggunakan template yang divalidasi.

Rencanakan untuk menggunakan referensi CloudFormation template

Saat Anda menulis templat, Anda dapat menemukan dokumentasi untuk sintaks terperinci untuk berbagai jenis sumber daya dalam [referensi jenis AWS sumber daya dan properti](#).

Seringkali, templat tumpukan Anda akan memerlukan fungsi intrinsik untuk menetapkan nilai properti yang tidak tersedia hingga runtime dan atribut khusus untuk mengontrol perilaku sumber daya. Saat Anda menulis templat Anda, lihat sumber daya berikut untuk panduan:

- [Referensi fungsi intrinsik](#) — Beberapa fungsi intrinsik yang umum digunakan meliputi:
 - `Ref`— Mengambil nilai parameter atau ID fisik sumber daya.

- Sub— Mengganti placeholder dalam string dengan nilai aktual.
- GetAtt— Mengembalikan nilai atribut dari sumber daya dalam template.
- Join— Bergabung dengan satu set nilai ke dalam satu string.
- [Referensi atribut sumber daya](#) - Beberapa atribut khusus yang umum digunakan meliputi:
 - DependsOn— Gunakan atribut ini untuk menentukan bahwa satu sumber daya harus dibuat setelah yang lain.
 - DeletionPolicy— Gunakan atribut ini untuk menentukan bagaimana CloudFormation seharusnya menangani penghapusan sumber daya.

Sampel template

CloudFormation menyediakan template tumpukan sumber terbuka yang dapat Anda gunakan untuk memulai. Untuk informasi selengkapnya, lihat [AWS CloudFormation Contoh Template](#) di GitHub situs web.

Perlu diingat bahwa template ini tidak dimaksudkan untuk siap produksi. Anda harus meluangkan waktu untuk mempelajari cara kerjanya, menyesuaikannya dengan kebutuhan Anda, dan memastikan bahwa mereka memenuhi standar kepatuhan perusahaan Anda.

Setiap template dalam repositori ini melewati pemeriksaan [CloudFormation Linter](#) (cfn-lint), dan juga seperangkat AWS CloudFormation Guard aturan dasar berdasarkan 20 Center for Internet Security (CIS) Teratas, dengan pengecualian untuk beberapa aturan yang masuk akal untuk menjaga sampel tetap fokus pada satu kasus penggunaan.

CloudFormation format templat

Anda dapat membuat CloudFormation template dalam format JSON atau YAMAL. Kedua format melayani tujuan yang sama tetapi menawarkan keuntungan yang berbeda dalam hal keterbacaan dan kompleksitas.

- JSON - JSON adalah format pertukaran data ringan yang mudah diurai dan dihasilkan oleh mesin. Namun, dapat menjadi rumit bagi manusia untuk membaca dan menulis, terutama untuk konfigurasi yang kompleks. Di JSON, template disusun menggunakan tanda kurung {} dan tanda kurung bersarang [] untuk menentukan sumber daya, parameter, dan komponen lainnya. Sintaksnya membutuhkan deklarasi eksplisit dari setiap elemen, yang dapat membuat template bertele-tele tetapi memastikan kepatuhan yang ketat terhadap format terstruktur.

- YAMAL — YAMAL dirancang agar lebih mudah dibaca manusia dan kurang bertele-tele daripada JSON. Ini menggunakan lekukan daripada kawat gigi dan tanda kurung untuk menunjukkan bersarang, yang dapat membuatnya lebih mudah untuk memvisualisasikan hierarki sumber daya dan parameter. YAMAL sering disukai karena kejelasan dan kemudahan penggunaannya, terutama ketika berhadapan dengan template yang lebih kompleks. Namun, ketergantungan YAMAL pada lekukan dapat menyebabkan kesalahan jika jarak tidak konsisten, yang memerlukan perhatian yang cermat untuk menjaga akurasi.

Struktur templat

CloudFormation template dibagi menjadi beberapa bagian yang berbeda, dan setiap bagian dirancang untuk menyimpan jenis informasi tertentu. Beberapa bagian harus dinyatakan dalam urutan tertentu, dan untuk yang lain, urutannya tidak masalah. Namun, saat Anda membangun template Anda, akan sangat membantu untuk menggunakan urutan logis yang ditunjukkan dalam contoh berikut karena nilai dalam satu bagian mungkin merujuk ke nilai dari bagian sebelumnya.

Saat membuat template, jangan gunakan duplikat bagian utama, seperti bagian. `Resources` Meskipun CloudFormation mungkin menerima template, itu akan memiliki perilaku yang tidak terdefinisi saat memproses template, dan mungkin salah menyediakan sumber daya, atau mengembalikan kesalahan yang tidak dapat dijelaskan.

JSON

Contoh berikut menunjukkan struktur template berformat JSON dengan semua bagian yang tersedia.

```
{
  "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

Contoh berikut menunjukkan struktur template yang diformat YAML dengan semua bagian yang tersedia.

```
---  
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*

Komentar

Dalam template berformat JSON, komentar tidak didukung. JSON, menurut desain, tidak menyertakan sintaks untuk komentar, yang berarti Anda tidak dapat menambahkan komentar secara langsung dalam struktur JSON. Namun, jika Anda perlu menyertakan catatan penjelasan atau dokumentasi, Anda dapat mempertimbangkan untuk menambahkan metadata. Untuk informasi lebih lanjut, lihat [Atribut Metadata](#).

Dalam template berformat YAML, Anda dapat menyertakan komentar sebaris dengan menggunakan simbol. #

Contoh berikut menunjukkan templat YAML dengan komentar sejajar.

```
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
```

Spesifikasi

CloudFormation mendukung spesifikasi JSON dan YAMAL berikut:

JSON

CloudFormation mengikuti standar ECMA-404 JSON. Untuk informasi tentang format JSON, lihat <http://www.json.org>.

YAML

CloudFormation mendukung spesifikasi YAMAL Versi 1.1 dengan beberapa pengecualian. CloudFormation tidak mendukung fitur berikut:

- Tanda binary, omap, pairs, set, dan timestamp
- Alias
- Penggabungan hash

Untuk informasi lebih lanjut tentang YAMAL, lihat <https://yaml.org>.

Pelajari selengkapnya

Untuk setiap sumber daya yang Anda tentukan dalam template, Anda menentukan properti dan nilainya menggunakan aturan sintaks tertentu dari JSON atau YAMAL. Untuk informasi lebih lanjut tentang sintaksis templat untuk setiap format, lihat [CloudFormation bagian template](#).

Menggunakan ekspresi reguler dalam CloudFormation template

[Anda dapat menggunakan ekspresi reguler \(umumnya dikenal sebagai regex\) di sejumlah tempat dalam CloudFormation template Anda, seperti untuk AllowedPattern properti saat membuat parameter template.](#)

Semua ekspresi reguler CloudFormation sesuai dengan sintaks regex Java. [Untuk deskripsi komprehensif tentang sintaks regex Java dan konstruksinya, lihat java.util.Regex.Pattern.](#)

Jika Anda menulis CloudFormation template Anda dalam sintaks JSON, Anda harus menghindari karakter garis miring terbalik (\) dalam ekspresi reguler Anda dengan menambahkan garis miring terbalik tambahan. Ini karena JSON menafsirkan garis miring terbalik sebagai karakter pelarian, dan Anda harus menghindarinya untuk memastikan mereka diperlakukan sebagai garis miring terbalik literal dalam ekspresi reguler.

Misalnya, jika Anda menyertakan \d dalam ekspresi reguler Anda untuk mencocokkan karakter digit, Anda perlu menulisnya sebagai \\d dalam templat JSON Anda.

Dalam contoh berikut, AllowedPattern properti menentukan ekspresi reguler yang cocok dengan empat karakter digit berturut-turut (). \d{4} Namun, karena ekspresi reguler didefinisikan dalam template JSON, karakter garis miring terbalik perlu diloloskan dengan garis miring terbalik tambahan (). \\d

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

Jika Anda menulis CloudFormation template Anda dalam sintaks YAML, Anda harus mengelilingi ekspresi reguler dengan tanda kutip tunggal ('). Tidak diperlukan pelarian tambahan.

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

CloudFormation bagian template

Setiap CloudFormation template terdiri dari satu atau lebih bagian, masing-masing melayani tujuan tertentu.

Bagian Sumber Daya diperlukan di setiap CloudFormation template dan membentuk inti dari template. Bagian ini menentukan sumber daya tumpukan dan propertinya, seperti EC2 instans Amazon atau bucket Amazon S3. Setiap sumber daya didefinisikan dengan ID logis, tipe, dan detail konfigurasi spesifik yang unik.

Bagian Parameter, meskipun opsional, memainkan peran penting dalam membuat template lebih fleksibel. Ini memungkinkan pengguna untuk meneruskan nilai saat runtime saat membuat atau memperbarui tumpukan. Parameter ini dapat direferensikan di Outputs bagian Resources dan, memungkinkan kustomisasi tanpa mengubah template itu sendiri. Misalnya, Anda dapat menggunakan parameter untuk menentukan jenis instans atau pengaturan lingkungan yang bervariasi di antara penerapan.

Bagian Output, juga opsional, mendefinisikan nilai yang dikembalikan saat melihat properti tumpukan. Output memberikan informasi yang berguna seperti pengidentifikasi sumber daya atau URLs, yang dapat dimanfaatkan untuk tujuan operasional atau untuk integrasi dengan tumpukan lain. Bagian ini membantu pengguna mengambil dan menggunakan detail penting tentang sumber daya yang dibuat oleh template.

Bagian opsional lainnya termasuk Pemetaan, yang berfungsi seperti tabel pencarian untuk mengelola nilai bersyarat. Dengan pemetaan, Anda menentukan pasangan kunci-nilai dan menggunakannya dengan fungsi `Fn::FindInMap` intrinsik di bagian `Resources` Outputs. Ini berguna untuk skenario di mana Anda perlu menyesuaikan konfigurasi berdasarkan kondisi seperti Wilayah AWS atau lingkungan.

Bagian Metadata dan Aturan, meskipun kurang umum digunakan, menyediakan fungsionalitas tambahan. Metadata dapat menyertakan informasi tambahan tentang template, sementara Rules memvalidasi parameter atau kombinasi parameter selama pembuatan tumpukan atau pembaruan, memastikan mereka memenuhi kriteria tertentu. Bagian Kondisi lebih lanjut meningkatkan fleksibilitas dengan mengontrol apakah sumber daya tertentu dibuat atau properti diberi nilai berdasarkan kondisi seperti tipe lingkungan.

Terakhir, bagian Transform digunakan untuk menerapkan makro selama pemrosesan template. Untuk aplikasi tanpa server (juga disebut sebagai aplikasi Lambda), ini menentukan versi Model [Aplikasi AWS Tanpa Server](#) (SAM) yang akan digunakan. AWS Ketika Anda menentukan transformasi, Anda dapat menggunakan AWS SAM sintaks untuk mendeklarasikan sumber daya dalam template Anda. Model mendefinisikan sintaks yang dapat Anda gunakan dan bagaimana itu diproses. Anda juga dapat menggunakan `AWS::Include` transformasi untuk menyertakan cuplikan template yang disimpan secara terpisah dari template utama CloudFormation.

Topik berikut memberikan lebih banyak informasi dan contoh untuk menggunakan setiap bagian.

Topik

- [CloudFormation Resourcesintaks template](#)

- [CloudFormation Parameterssintaks template](#)
- [CloudFormation Outputssintaks template](#)
- [CloudFormation Mappingssintaks template](#)
- [CloudFormation Metadatasintaks template](#)
- [CloudFormation Rulessintaks template](#)
- [CloudFormation Conditionssintaks template](#)
- [CloudFormation Transformbagian template](#)
- [CloudFormation format template sintaks versi](#)
- [CloudFormation template Description sintaksis](#)

CloudFormation Resourcessintaks template

ResourcesBagian ini adalah bagian tingkat atas yang diperlukan dalam CloudFormation templat. Ini mendeklarasikan AWS sumber daya yang CloudFormation ingin Anda sediakan dan konfigurasi sebagai bagian dari tumpukan Anda.

Sintaksis

ResourcesBagian ini menggunakan sintaks berikut:

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
      ...
```

Logical ID (juga disebut nama logis)

Dalam CloudFormation template, sumber daya diidentifikasi dengan nama sumber daya logisnya. Nama-nama ini harus alfanumerik (A-za-Z0-9) dan unik dalam template. Nama logis digunakan untuk referensi sumber daya dari bagian lain dari template.

Jenis sumber daya

Setiap sumber daya harus memiliki Type atribut, yang mendefinisikan jenis AWS sumber daya itu. TypeAtribut memiliki format `AWS::ServiceName::ResourceType`. Misalnya, Type atribut untuk bucket Amazon S3 adalah `AWS::S3::Bucket`

Untuk daftar lengkap jenis sumber daya yang didukung, lihat [referensi jenis AWS sumber daya dan properti](#).

Properti sumber daya

Properti sumber daya adalah opsi tambahan yang dapat Anda tentukan untuk menentukan detail konfigurasi untuk jenis sumber daya tertentu. Beberapa properti diperlukan, sementara yang lain opsional. Beberapa properti memiliki nilai default, sehingga menentukan properti tersebut adalah opsional.

Untuk detail tentang properti yang didukung untuk setiap jenis sumber daya, lihat topik dalam [referensi tipe AWS sumber daya dan properti](#).

Nilai properti dapat berupa string literal, daftar string, Boolean, referensi dinamis, referensi parameter, referensi semu, atau nilai yang dikembalikan oleh fungsi. Contoh berikut menunjukkan cara mendeklarasikan jenis nilai properti yang berbeda:

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
```

ID Fisik

Selain ID logis, sumber daya tertentu juga memiliki ID fisik, yang merupakan nama aktual yang ditetapkan untuk sumber daya tersebut, seperti ID EC2 instance atau nama bucket S3. Gunakan fisik IDs untuk mengidentifikasi sumber daya di luar CloudFormation templat, tetapi hanya setelah sumber daya dibuat. Misalnya, Anda memberikan sumber daya EC2 instance ID logis dari `MyEC2Instance`. Saat CloudFormation membuat instance, CloudFormation secara otomatis menghasilkan dan menetapkan ID fisik (seperti `i-1234567890abcdef0`) ke instance. Anda dapat menggunakan ID fisik ini untuk mengidentifikasi instance dan melihat propertinya (seperti nama DNS) dengan menggunakan EC2 konsol Amazon.

Untuk bucket Amazon S3 dan banyak sumber daya lainnya, CloudFormation secara otomatis menghasilkan nama fisik unik untuk sumber daya jika Anda tidak menentukannya secara eksplisit. Nama fisik ini didasarkan pada kombinasi nama CloudFormation tumpukan, nama logis sumber daya yang ditentukan dalam CloudFormation template, dan ID unik. Misalnya, jika Anda memiliki bucket Amazon S3 dengan nama logis `MyBucket` dalam tumpukan bernama `MyStack`, CloudFormation beri nama bucket dengan nama fisik berikut. `MyStack-MyBucket-abcdefghijkl`

Untuk sumber daya yang mendukung nama kustom, Anda dapat menetapkan nama fisik Anda sendiri untuk membantu Anda mengidentifikasi sumber daya dengan cepat. Misalnya, Anda dapat menamakan bucket S3 yang menyimpan log sebagai `MyPerformanceLogs`. Untuk informasi lebih lanjut, lihat [Jenis nama](#).

Referensi sumber daya

Seringkali, Anda perlu mengatur properti pada satu sumber daya berdasarkan nama atau properti sumber daya lain. Misalnya, Anda dapat membuat EC2 instance yang menggunakan grup EC2 keamanan, atau CloudFront distribusi yang didukung oleh bucket S3. Semua sumber daya ini dapat dibuat dalam CloudFormation template yang sama.

CloudFormation menyediakan fungsi intrinsik yang dapat Anda gunakan untuk merujuk ke sumber daya lain dan propertinya. Fungsi-fungsi ini memungkinkan Anda untuk membuat dependensi antara sumber daya dan meneruskan nilai dari satu sumber daya ke sumber daya lainnya.

Fungsi **Ref**

`Ref` fungsi ini biasanya digunakan untuk mengambil properti identifikasi sumber daya yang didefinisikan dalam CloudFormation template yang sama. Apa yang dikembalikan tergantung pada jenis sumber daya. Untuk sebagian besar sumber daya, ia mengembalikan nama fisik sumber daya. Namun, untuk beberapa jenis sumber daya, mungkin mengembalikan nilai yang berbeda, seperti alamat IP untuk `AWS::EC2::EIP` sumber daya atau Nama Sumber Daya Amazon (ARN) untuk topik Amazon SNS.

Contoh berikut menunjukkan bagaimana menggunakan `Ref` fungsi dalam properti. Dalam setiap contoh ini, `Ref` fungsi akan mengembalikan nama sebenarnya dari `LogicalResourceName` sumber daya yang dideklarasikan di tempat lain dalam template. Contoh `!Ref` sintaks dalam contoh YAMAL hanyalah cara yang lebih pendek untuk menulis fungsi. `Ref`

JSON

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

YAML

```
Properties:
```

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

Untuk informasi lebih rinci tentang Ref fungsi, lihat [Ref](#).

Fungsi **Fn::GetAtt**

RefFungsi ini membantu jika parameter atau nilai yang dikembalikan untuk sumber daya persis seperti yang Anda inginkan. Namun, Anda mungkin memerlukan atribut lain dari sumber daya. Misalnya, jika Anda ingin membuat CloudFront distribusi dengan asal S3, Anda perlu menentukan lokasi bucket dengan menggunakan alamat gaya DNS. Sejumlah sumber daya memiliki atribut tambahan yang nilainya dapat Anda gunakan di templat Anda. Untuk mendapatkan atribut ini, Anda menggunakan Fn::GetAtt fungsi.

Contoh berikut menunjukkan bagaimana menggunakan GetAtt fungsi dalam properti.

Fn::GetAttFungsi ini mengambil dua parameter, nama logis sumber daya dan nama atribut yang akan diambil. Contoh !GetAtt sintaks dalam contoh YAMAL hanyalah cara yang lebih pendek untuk menulis fungsi. GetAtt

JSON

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

YAML

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

Untuk informasi lebih rinci tentang GetAtt fungsi, lihat [Fn::GetAtt](#).

Contoh

Contoh berikut menggambarkan cara mendeklarasikan sumber daya dan bagaimana CloudFormation template dapat mereferensikan sumber daya lain yang ditentukan dalam template yang sama dan sumber daya yang ada. AWS

Topik

- [Mendeklarasikan sumber daya tunggal dengan nama kustom](#)
- [Mereferensikan sumber daya lain dengan fungsi Ref](#)
- [Mereferensikan atribut sumber daya menggunakan fungsi Fn::GetAtt](#)

Mendeklarasikan sumber daya tunggal dengan nama kustom

Contoh berikut mendeklarasikan sumber daya tunggal tipe `AWS::S3::Bucket` dengan nama logis. `MyBucket` `BucketName` properti diatur ke `amzn-s3-demo-bucket`, yang harus diganti dengan nama yang diinginkan untuk bucket S3 Anda.

Jika Anda menggunakan deklarasi sumber daya ini untuk membuat tumpukan, CloudFormation akan membuat bucket Amazon S3 dengan pengaturan default. Untuk sumber daya lain, seperti EC2 instans Amazon atau grup Auto Scaling, CloudFormation memerlukan informasi lebih lanjut.

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***Mereferensikan sumber daya lain dengan fungsi Ref**

Contoh berikut menunjukkan deklarasi sumber daya yang mendefinisikan EC2 instance dan grup keamanan. Ec2InstanceSumber daya mereferensikan InstanceSecurityGroup sumber daya sebagai bagian dari SecurityGroupIds propertinya menggunakan Ref fungsi. Ini juga mencakup grup keamanan yang ada (sg-12a4c434) yang tidak dideklarasikan dalam template. Anda menggunakan string literal untuk mereferensikan ke sumber daya AWS yang ada.

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
```

Mereferensikan atribut sumber daya menggunakan fungsi **Fn::GetAtt**

Contoh berikut menunjukkan deklarasi sumber daya yang mendefinisikan sumber daya CloudFront distribusi dan bucket S3. `MyDistribution` sumber daya menentukan nama DNS `MyBucket` sumber daya menggunakan `Fn::GetAtt` fungsi untuk mendapatkan atribut bucket. `DomainName` Anda akan melihat bahwa `Fn::GetAtt` fungsi tersebut mencantumkan dua parameternya dalam sebuah array. Untuk fungsi yang mengambil beberapa parameter, gunakan array untuk menentukan parameternya.

JSON

```
{
  "Resources": {
    "MyBucket": {
      "Type": "AWS::S3::Bucket"
    },
    "MyDistribution": {
      "Type": "AWS::CloudFront::Distribution",
      "Properties": {
```

```

    "DistributionConfig": {
      "Origins": [
        {
          "DomainName": {
            "Fn::GetAtt": [
              "MyBucket",
              "DomainName"
            ]
          },
          "Id": "MyS3Origin",
          "S3OriginConfig": {}
        }
      ],
      "Enabled": "true",
      "DefaultCacheBehavior": {
        "TargetOriginId": "MyS3Origin",
        "ForwardedValues": {
          "QueryString": "false"
        },
        "ViewerProtocolPolicy": "allow-all"
      }
    }
  }
}

```

YAML

```

Resources:
  MyBucket:
    Type: 'AWS::S3::Bucket'
  MyDistribution:
    Type: 'AWS::CloudFront::Distribution'
    Properties:
      DistributionConfig:
        Origins:
          - DomainName: !GetAtt
            MyBucket
            DomainName
          Id: MyS3Origin
          S3OriginConfig: {}
        Enabled: 'true'

```

```
DefaultCacheBehavior:
  TargetOriginId: MyS3Origin
  ForwardedValues:
    QueryString: 'false'
  ViewerProtocolPolicy: allow-all
```

CloudFormation Parameters sintaks template

Gunakan bagian Parameters untuk menyesuaikan templat Anda. Dengan parameter, Anda dapat memasukkan nilai kustom ke template Anda setiap kali Anda membuat atau memperbarui tumpukan. Dengan menggunakan parameter dalam template Anda, Anda dapat membuat template yang dapat digunakan kembali dan fleksibel yang dapat disesuaikan dengan skenario tertentu.

Dengan menentukan parameter dari jenis yang sesuai, Anda dapat memilih dari daftar pengidentifikasi sumber daya yang ada saat Anda menggunakan konsol untuk membuat tumpukan Anda. Untuk informasi selengkapnya, lihat [Tentukan sumber daya yang ada saat runtime dengan CloudFormation tipe parameter yang disediakan](#).

Parameter adalah cara populer untuk menentukan nilai properti sumber daya tumpukan. Namun, mungkin ada pengaturan yang bergantung pada wilayah atau agak rumit bagi pengguna untuk mencari tahu karena kondisi atau dependensi lain. Dalam kasus ini, Anda mungkin ingin menempatkan beberapa logika dalam template itu sendiri sehingga pengguna dapat menentukan nilai yang lebih sederhana (atau tidak sama sekali) untuk mendapatkan hasil yang mereka inginkan, seperti dengan menggunakan pemetaan. Untuk informasi selengkapnya, lihat [CloudFormation Mapping sintaks template](#).

Sintaksis

Anda mendeklarasikan parameter di Parameters bagian template, yang menggunakan sintaks umum berikut:

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
```

Parameter berisi daftar atribut yang menentukan nilai dan batasan terhadap nilainya. Satu-satunya atribut yang diperlukan adalah `Type`, yang dapat berupa `StringNumber`, atau tipe parameter yang CloudFormation disediakan. Anda juga dapat menambahkan `Description` atribut yang menjelaskan jenis nilai apa yang harus Anda tentukan. Nama dan deskripsi parameter muncul di halaman Tentukan Parameter saat Anda menggunakan templat di wizard Buat Tumpukan.

Note

Secara default, CloudFormation konsol mencantumkan parameter input menurut abjad berdasarkan ID logisnya. Untuk mengganti urutan default ini dan mengelompokkan parameter terkait bersama-sama, Anda dapat menggunakan kunci `AWS::CloudFormation::Interface` metadata di template Anda. Untuk informasi selengkapnya, lihat [Mengatur CloudFormation parameter dengan AWS::CloudFormation::Interface metadata](#).

Untuk parameter dengan nilai default, CloudFormation gunakan nilai default kecuali pengguna menentukan nilai lain. Jika Anda menghilangkan atribut default, pengguna diminta untuk menentukan nilai untuk parameter tersebut. Namun, mengharuskan pengguna untuk memasukkan nilai tidak memastikan bahwa nilainya valid. Untuk memvalidasi nilai parameter, Anda dapat mendeklarasikan batasan atau menentukan tipe parameter -spesifik. AWS

Untuk parameter tanpa nilai default, pengguna harus menentukan nilai nama kunci pada pembuatan tumpukan. Jika tidak, CloudFormation gagal membuat tumpukan dan melempar pengecualian:

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

Properti

AllowedPattern

Ekspresi reguler yang mewakili pola untuk memungkinkan String atau CommaDelimitedList mengetik. Ketika diterapkan pada parameter tipeString, pola harus cocok dengan seluruh nilai parameter yang disediakan. Ketika diterapkan ke parameter tipeCommaDelimitedList, pola harus cocok dengan setiap nilai dalam daftar.

Wajib: Tidak

AllowedValues

Array yang berisi daftar nilai yang diizinkan untuk parameter. Ketika diterapkan ke parameter tipeString, nilai parameter harus menjadi salah satu nilai yang diizinkan. Ketika diterapkan ke parameter tipeCommaDelimitedList, setiap nilai dalam daftar harus menjadi salah satu nilai yang diizinkan yang ditentukan.

Wajib: Tidak

Note

Jika Anda menggunakan YAMAL dan ingin menggunakan Yes dan No string untukAllowedValues, gunakan tanda kutip tunggal untuk mencegah parser YAMAL mempertimbangkan nilai boolean ini.

ConstraintDescription

Suatu string yang menjelaskan kendala ketika ia dilanggar. Sebagai contoh, tanpa deskripsi kendala, parameter yang memiliki pola yang diizinkan [A-Za-z0-9]+ menampilkan pesan kesalahan berikut ketika pengguna menentukan nilai yang tidak valid:

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

Dengan menambahkan deskripsi kendala, seperti hanya boleh berisi huruf (huruf besar dan huruf kecil) dan angka, Anda dapat menampilkan pesan kesalahan yang disesuaikan berikut:

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

Wajib: Tidak

Default

Nilai tipe yang sesuai untuk templat untuk digunakan jika tidak ada nilai yang ditentukan ketika tumpukan dibuat. Jika Anda menentukan kendala untuk parameter, Anda harus menentukan nilai yang mematuhi kendala tersebut.

Wajib: Tidak

Description

String hingga 4000 karakter yang menggambarkan parameter.

Wajib: Tidak

MaxLength

Nilai integer yang menentukan jumlah karakter terbesar dengan `String` tipe yang Anda izinkan.

Wajib: Tidak

MaxValue

Nilai numerik yang menentukan nilai numerik terbesar dengan `Number` tipe yang ingin Anda izinkan.

Wajib: Tidak

MinLength

Nilai integer yang menentukan jumlah karakter terbesar dengan `String` tipe yang ingin Anda izinkan.

Wajib: Tidak

MinValue

Nilai numerik yang menentukan nilai numerik terkecil dengan `Number` tipe yang ingin Anda izinkan.

Wajib: Tidak

NoEcho

Apakah akan menutupi nilai parameter untuk mencegahnya ditampilkan di konsol, alat baris perintah, atau API. Jika Anda menyetel `NoEcho` atribut `true`, CloudFormation mengembalikan

nilai parameter yang disamarkan sebagai tanda bintang (*****) untuk setiap panggilan yang menggambarkan peristiwa tumpukan atau tumpukan, kecuali untuk informasi yang disimpan di lokasi yang ditentukan di bawah ini.

Wajib: Tidak

 Important

Menggunakan NoEcho atribut tidak menutupi informasi apa pun yang disimpan sebagai berikut:

- Bagian Metadata template. CloudFormation tidak mengubah, memodifikasi, atau menyunting informasi apa pun yang Anda sertakan di Metadata bagian ini. Untuk informasi selengkapnya, lihat [Metadata](#).
- Bagian Outputs template. Untuk informasi selengkapnya, lihat [Output](#).
- MetadataAtribut definisi sumber daya. Untuk informasi lebih lanjut, lihat [Atribut Metadata](#).

Kami sangat menyarankan Anda untuk tidak menggunakan mekanisme ini untuk memasukkan informasi sensitif, seperti kata sandi atau rahasia.

 Important

Daripada menyimpan informasi sensitif secara langsung di CloudFormation template Anda, kami sarankan Anda menggunakan parameter dinamis dalam template tumpukan untuk mereferensikan informasi sensitif yang disimpan dan dikelola di luar CloudFormation, seperti di AWS Systems Manager Parameter Store atau AWS Secrets Manager.

Untuk informasi selengkapnya, lihat [Do not embed credentials in your templates](#) best practice.

 Important

Kami sangat menyarankan agar tidak termasuk NoEcho parameter, atau data sensitif apa pun, di properti sumber daya yang merupakan bagian dari pengenalan utama sumber daya. Ketika NoEcho parameter disertakan dalam properti yang membentuk pengenalan sumber daya utama, CloudFormation dapat menggunakan nilai plaintext aktual dalam

pengidentifikasi sumber daya utama. ID sumber daya ini mungkin muncul dalam output turunan atau tujuan.

Untuk menentukan properti sumber daya mana yang terdiri dari pengidentifikasi utama tipe sumber daya, lihat dokumentasi referensi sumber daya untuk sumber daya tersebut dalam referensi [tipe AWS sumber daya dan properti](#). Di bagian Kembalikan nilai, nilai Ref pengembalian fungsi mewakili properti sumber daya yang terdiri dari tipe pengidentifikasi utama sumber daya.

Type

Tipe data untuk parameter (DataType).

Wajib: Ya

CloudFormation mendukung jenis parameter berikut:

String

Sebuah string literal. Anda dapat menggunakan atribut berikut untuk mendeklarasikan batasan:MinLength,,, MaxLength dan. Default AllowedValues AllowedPattern

Misalnya, pengguna dapat menentukan "MyUserName".

Number

Sebuah integer atau float. CloudFormation memvalidasi nilai parameter sebagai angka; Namun, ketika Anda menggunakan parameter di tempat lain di template Anda (misalnya, dengan menggunakan fungsi Ref intrinsik), nilai parameter menjadi string.

Anda dapat menggunakan atribut berikut untuk mendeklarasikan batasan:MinValue,,, dan. MaxValue Default AllowedValues

Misalnya, pengguna dapat menentukan "8888".

List<Number>

Array integer atau float yang dipisahkan oleh koma. CloudFormation memvalidasi nilai parameter sebagai angka; namun, ketika Anda menggunakan parameter di tempat lain dalam templat Anda (misalnya, dengan menggunakan Ref fungsi intrinsik), nilai parameter menjadi daftar string.

Misalnya, pengguna dapat menentukan "80,20", dan Ref akan menghasilkan ["80", "20"].

CommaDelimitedList

Array string literal yang dipisahkan dengan koma. Jumlah total string harus satu lebih dari jumlah koma. Juga, setiap string anggota dipangkas ruang.

Misalnya, pengguna dapat menentukan "test,dev,prod", dan Ref akan menghasilkan ["test", "dev", "prod"].

AWS-jenis parameter tertentu

AWS nilai-nilai seperti nama EC2 key pair Amazon dan VPC IDs. Untuk informasi selengkapnya, lihat [Tentukan sumber daya yang ada saat runtime](#).

Jenis parameter Systems Manager

Parameter yang sesuai dengan parameter yang ada di Systems Manager Parameter Store. Anda menentukan kunci parameter Systems Manager sebagai nilai tipe parameter Systems Manager, dan CloudFormation mengambil nilai terbaru dari Parameter Store yang akan digunakan untuk tumpukan. Untuk informasi selengkapnya, lihat [Tentukan sumber daya yang ada saat runtime](#).

Persyaratan umum untuk parameter

Persyaratan berikut berlaku saat menggunakan parameter:

- Anda dapat memiliki maksimum 200 parameter dalam CloudFormation template.
- Setiap parameter harus diberi nama logis (juga disebut ID logis) yang harus alfanumerik dan unik di antara semua nama logis dalam template.
- Setiap parameter harus diberi tipe parameter yang didukung oleh CloudFormation. Untuk informasi lebih lanjut, lihat [Tipe](#).
- Setiap parameter harus diberi nilai saat runtime CloudFormation agar berhasil menyediakan tumpukan. Anda dapat secara opsional menentukan nilai default CloudFormation untuk digunakan kecuali nilai lain disediakan.
- Parameter harus dinyatakan dan direferensikan dari dalam templat yang sama. Anda dapat mereferensikan parameter dari bagian Resources dan Outputs templat tersebut.

Contoh

Topik

- [Parameter string sederhana](#)
- [Parameter kata sandi](#)
- [Parameter referensi](#)
- [Parameter daftar yang dibatasi koma](#)
- [Kembalikan nilai dari parameter daftar yang dipisahkan koma](#)

Parameter string sederhana

Contoh berikut mendeklarasikan parameter bernama `InstanceTypeParameter` tipe `String`. Parameter ini memungkinkan Anda menentukan jenis EC2 instans Amazon untuk tumpukan. Jika tidak ada nilai yang diberikan selama pembuatan atau pembaruan tumpukan, CloudFormation gunakan nilai default `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 kata sandi

Contoh berikut mendeklarasikan parameter bernama DBPwd tipe `String` tanpa nilai default. `NoEcho` properti diatur `true` untuk mencegah nilai parameter ditampilkan dalam deskripsi tumpukan. Nilai minimum yang dapat ditentukan adalah 1, dan panjang maksimum yang dapat ditentukan adalah 41. Pola ini memungkinkan huruf kecil dan huruf besar karakter abjad dan angka. Contoh ini juga menggambarkan penggunaan ekspresi reguler untuk `AllowedPattern` properti.

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 referensi

Anda menggunakan `Ref` fungsi intrinsik untuk referensi parameter, dan CloudFormation menggunakan nilai parameter untuk penyediaan tumpukan. Anda dapat mereferensikan parameter dari bagian `Resources` dan `Outputs` dari templat yang sama.

Dalam contoh berikut, `InstanceType` properti sumber daya EC2 contoh mereferensikan nilai `InstanceTypeParameter` parameter:

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
```

Parameter daftar yang dibatasi koma

Jenis `CommaDelimitedList` parameter dapat berguna ketika Anda perlu memberikan beberapa nilai untuk satu properti. Contoh berikut mendeklarasikan parameter bernama `DbSubnetIpBlocks` dengan nilai default dari tiga blok CIDR dipisahkan dengan koma.

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"
```

Kembalikan nilai dari parameter daftar yang dipisahkan koma

Untuk merujuk ke nilai tertentu dalam daftar parameter yang dibatasi koma, gunakan fungsi `Fn::Select` intrinsik di bagian templat Anda. Resources Lulus nilai indeks objek yang Anda inginkan dan daftar objek, seperti yang ditunjukkan pada contoh berikut.

JSON

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

```

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

```

```

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

```

YAML

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

```

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

```

Sumber daya terkait

CloudFormation juga mendukung penggunaan referensi dinamis untuk menentukan nilai properti secara dinamis. Misalnya, Anda mungkin perlu mereferensikan string aman yang disimpan di Systems Manager Parameter Store. Untuk informasi selengkapnya, lihat [Dapatkan nilai yang disimpan di layanan lain menggunakan referensi dinamis](#).

Anda juga dapat menggunakan parameter semu dalam fungsi Ref atau Sub fungsi untuk mengisi nilai secara dinamis. Lihat informasi yang lebih lengkap di [Dapatkan AWS nilai menggunakan parameter semu](#).

CloudFormation Outputs sintaks template

Outputs Bagian opsional mendeklarasikan nilai output untuk tumpukan. Nilai output ini dapat digunakan dengan berbagai cara:

- Tangkap detail penting tentang sumber daya Anda — Output adalah cara mudah untuk menangkap informasi penting tentang sumber daya Anda. Misalnya, Anda dapat menampilkan nama bucket S3 untuk tumpukan agar bucket lebih mudah ditemukan. Anda dapat melihat nilai output di tab Output CloudFormation konsol atau dengan menggunakan perintah [describe-stacks](#) CLI.
- Referensi cross-stack — Anda dapat mengimpor nilai output ke tumpukan lain untuk [membuat referensi antar](#) tumpukan. Ini sangat membantu ketika Anda perlu berbagi sumber daya atau konfigurasi di beberapa tumpukan.

Important

CloudFormation tidak menyunting atau mengaburkan informasi apa pun yang Anda sertakan di bagian ini. Outputs Kami sangat merekomendasikan Anda untuk tidak menggunakan bagian ini untuk menampilkan informasi sensitif, seperti kata sandi atau rahasia. Nilai output tersedia setelah operasi tumpukan selesai. Nilai keluaran tumpukan tidak tersedia saat status tumpukan berada di salah satu IN_PROGRESS [status](#). Kami tidak merekomendasikan membuat dependensi antara waktu aktif layanan dan nilai output tumpukan karena nilai output mungkin tidak tersedia setiap saat.

Sintaksis

Bagian Outputs terdiri dari nama kunci Outputs. Anda dapat mendeklarasikan maksimum 200 output dalam sebuah templat.

Contoh berikut menunjukkan struktur bagian Outputs.

JSON

Gunakan tanda kurung untuk menutup seluruh deklarasi output. Membatasi beberapa output dengan koma.

```
"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
```

Bidang output

Bagian Outputs dapat mencakup bidang berikut.

Logical ID (juga disebut nama logis)

Sebuah pengidentifikasi untuk output saat ini. ID logis harus berupa alfanumerik (a–z, A–Z, 0–9) dan unik dalam templat.

Description (opsional)

Jenis String yang menjelaskan nilai output. Nilai untuk deklarasi deskripsi harus berupa string literal yang panjangnya antara 0 dan 1024 byte. Anda tidak dapat menggunakan parameter atau fungsi untuk menentukan deskripsi.

Value(diperlukan)

Nilai properti yang dikembalikan oleh perintah [describe-stacks](#). Nilai output dapat mencakup literal, referensi parameter, parameter semu, nilai pemetaan, atau fungsi intrinsik.

Export (opsional)

Nama output sumber daya yang akan diekspor untuk Referensi lintas tumpukan.

Anda dapat menggunakan fungsi intrinsik untuk menyesuaikan nilai Name ekspor.

Untuk informasi selengkapnya, lihat [Dapatkan output yang diekspor dari tumpukan yang diterapkan CloudFormation](#).

Untuk mengasosiasikan syarat dengan output, tentukan syarat di bagian [Conditions](#) dari templat.

Contoh

Contoh berikut menggambarkan cara kerja output tumpukan.

Topik

- [Output tumpukan](#)
- [Sesuaikan nama ekspor menggunakan Fn::Sub](#)
- [Sesuaikan nama ekspor menggunakan Fn::Join](#)
- [Kembalikan URL yang dibuat menggunakan Fn::Join](#)

Output tumpukan

Dalam contoh berikut, output bernama BackupLoadBalancerDNSName mengembalikan nama DNS untuk sumber daya dengan ID logis BackupLoadBalancer hanya ketika kondisi CreateProdResources benar. Output bernama InstanceID mengembalikan ID dari EC2 contoh dengan ID logisEC2Instance.

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
```

Sesuaikan nama ekspor menggunakan **Fn::Sub**

Dalam contoh berikut, output bernama StackVPC mengembalikan ID dari VPC, dan kemudian mengekspor nilai untuk referensi lintas tumpukan dengan nama VPCID ditambahkan ke nama tumpukan.

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"
```

Untuk informasi selengkapnya tentang fungsi Fn::Sub, lihat [Fn::Sub](#).

Sesuaikan nama ekspor menggunakan **Fn::Join**

Anda juga dapat menggunakan Fn::Join fungsi untuk membangun nilai berdasarkan parameter, atribut sumber daya, dan string lainnya.

Contoh berikut menggunakan Fn::Join fungsi untuk menyesuaikan nama ekspor alih-alih Fn::Sub fungsi. Fn::JoinFungsi contoh menggabungkan nama tumpukan dengan nama VPCID menggunakan titik dua sebagai pemisah.

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 ] ]
```

Untuk informasi selengkapnya tentang fungsi `Fn::Join`, lihat [Fn::Join](#).

Kembalikan URL yang dibuat menggunakan **Fn::Join**

Dalam contoh berikut untuk template yang membuat WordPress situs, `InstallURL` adalah string yang dikembalikan oleh panggilan `Fn::Join` fungsi yang menggabungkan `http://`, nama DNS sumber daya, dan `ElasticLoadBalancer /wp-admin/install.php`. Nilai output akan serupa dengan berikut ini:

```
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

```

Untuk informasi selengkapnya tentang fungsi `Fn::Join`, lihat [Fn::Join](#).

CloudFormation Mappingsintaks template

MappingsBagian opsional membantu Anda membuat pasangan kunci-nilai yang dapat digunakan untuk menentukan nilai berdasarkan kondisi atau dependensi tertentu.

Salah satu kasus penggunaan umum untuk Mappings bagian ini adalah untuk menetapkan nilai berdasarkan di Wilayah AWS mana tumpukan digunakan. Ini dapat dicapai dengan menggunakan parameter `AWS::Region` semu. Parameter `AWS::Region` pseudo adalah nilai yang CloudFormation menyelesaikan ke wilayah tempat tumpukan dibuat. Parameter pseudo diselesaikan CloudFormation saat Anda membuat tumpukan.

Untuk mengambil nilai dalam peta, Anda dapat menggunakan fungsi `Fn::FindInMap` intrinsik dalam Resources bagian template Anda.

Sintaksis

MappingsBagian ini menggunakan sintaks berikut:

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 adalah nama logis untuk pemetaan.
- Dalam pemetaan, setiap peta adalah kunci diikuti oleh pemetaan lain.
- Kunci harus berupa peta pasangan nama-nilai dan unik dalam pemetaan.
- Pasangan nama-nilai adalah label, dan nilai untuk memetakan. Dengan menamakan nilai-nilai, Anda dapat memetakan lebih dari satu set nilai ke kunci.
- Kunci dalam pemetaan harus berupa string literal.

- Nilainya bisa dari tipe `String` atau `List`.

Note

Anda tidak dapat menyertakan parameter, parameter semu, atau fungsi intrinsik dalam bagian Mappings.

Jika nilai dalam pemetaan saat ini tidak digunakan oleh tumpukan Anda, Anda tidak dapat memperbarui pemetaan saja. Anda harus menyertakan perubahan yang menambah, memodifikasi, atau menghapus sumber daya.

Contoh

Pemetaan Basic

Contoh berikut ini menunjukkan bagian Mappings dengan peta `RegionMap`, yang berisi lima kunci yang memetakan ke pasangan nama-nilai yang mengandung nilai-nilai string tunggal. Kunci adalah nama region. Setiap pasangan nama-nilai adalah tipe instance dari keluarga T yang tersedia di wilayah yang diwakili oleh kunci. Pasangan nama-nilai memiliki nama (`InstanceType` dalam contoh) dan nilai.

JSON

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

YAML

```
Mappings:
  RegionMap:
    us-east-1:
      InstanceType: t2.micro
    us-west-1:
```

```

    InstanceType: t2.micro
eu-west-1:
    InstanceType: t2.micro
eu-north-1:
    InstanceType: t3.micro
me-south-1:
    InstanceType: t3.micro

```

Pemetaan dengan beberapa nilai

Contoh berikut memiliki kunci wilayah yang dipetakan ke dua set nilai: satu bernama MyAMI1 dan yang lainnya MyAMI2.

JSON

```

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

```

YAML

```

AMIIDMap:
  us-east-1:
    MyAMI1: ami-0ff8a91507f77f867
    MyAMI2: ami-0a584ac55a7631c0c
  us-west-1:
    MyAMI1: ami-0bdb828fd58c52235
    MyAMI2: ami-066ee5fd4a9ef77f1
  eu-west-1:
    MyAMI1: ami-047bb4163c506cd98
    MyAMI2: ami-0a7c483d527806435
  ap-southeast-1:
    MyAMI1: ami-08569b978cc4dfa10
    MyAMI2: ami-0be9df32ae9f92309

```

```
ap-northeast-1:
  MyAMI1: ami-06cd52961ce9f0d85
  MyAMI2: ami-053cdd503598e4a9d
```

Mengembalikan nilai dari pemetaan

Anda dapat menggunakan fungsi `Fn::FindInMap` untuk mengembalikan nilai bernama berdasarkan kunci tertentu. Contoh template berikut berisi EC2 sumber daya Amazon yang `ImageId` propertinya ditetapkan oleh `FindInMap` fungsi. `FindInMap` fungsi menentukan kunci sebagai wilayah tempat tumpukan dibuat (menggunakan parameter `AWS::Region` semu) dan `MyAMI1` sebagai nama nilai untuk dipetakan. Untuk informasi selengkapnya tentang parameter semu, lihat [Dapatkan AWS nilai menggunakan parameter semu](#).

JSON

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

YAML

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

Parameter masukan dan **Fn::FindInMap**

Anda dapat menggunakan parameter input dengan fungsi `Fn::FindInMap` untuk merujuk ke nilai tertentu dalam peta. Misalnya, Anda memiliki daftar wilayah dan jenis lingkungan yang dipetakan ke ID grup keamanan tertentu. Anda dapat memilih ID grup keamanan yang digunakan stack Anda dengan menggunakan parameter input (`EnvironmentType`). Untuk menentukan wilayah, gunakan parameter `AWS::Region` semu, yang mendapatkan Wilayah AWS di mana Anda membuat tumpukan.

Contoh ini juga mendeklarasikan tipe parameter Systems Manager yang menyediakan parameter Systems Manager alias (`/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2`) sebagai nilai default untuk `ImageId` properti instance. EC2 Ini adalah nilai yang CloudFormation diselesaikan sebagai nilai ID AMI untuk Amazon Linux 2 AMI terbaru di wilayah tempat tumpukan dibuat.

JSON

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Parameters" : {
    "LatestAmiId" : {
      "Description" : "The latest Amazon Linux 2 AMI from the Parameter Store",
      "Type" : "AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>",
      "Default" : "/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2"
    },
    "EnvironmentType" : {
      "Description" : "The environment type (Dev or Prod)",
      "Type" : "String",
      "Default" : "Dev",
      "AllowedValues" : [
        "Dev",
        "Prod"
      ]
    }
  },
  "Mappings" : {
    "SecurityGroupMap" : {
      "us-east-1" : {
        "Dev" : "sg-12345678",
        "Prod" : "sg-abcdef01"
      },
      "us-west-2" : {
        "Dev" : "sg-ghijkl23",
        "Prod" : "sg-45678abc"
      }
    }
  },
  "Resources" : {
    "Ec2Instance" : {
      "Type" : "AWS::EC2::Instance",
      "Properties" : {
        "ImageId" : {
          "Ref" : "LatestAmiId"
        },
        "InstanceType" : "t2.micro",
        "SecurityGroupIds" : [{ "Fn::FindInMap" : [ "SecurityGroupMap", { "Ref" :
"AWS::Region" } ], { "Ref" : "EnvironmentType" } ]}]
      }
    }
  }
}
```

```
}  
}
```

YAML

```
AWS::TemplateFormatVersion: 2010-09-09  
Parameters:  
  LatestAmiId:  
    Description: The latest Amazon Linux 2 AMI from the Parameter Store  
    Type: 'AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>  
    Default: '/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2'  
  EnvironmentType:  
    Description: The environment type (Dev or Prod)  
    Type: String  
    Default: Dev  
    AllowedValues:  
      - Dev  
      - Prod  
Mappings:  
  SecurityGroupMap:  
    us-east-1:  
      Dev: "sg-12345678"  
      Prod: "sg-abcdef01"  
    us-west-2:  
      Dev: "sg-ghijkl23"  
      Prod: "sg-45678abc"  
Resources:  
  Ec2Instance:  
    Type: 'AWS::EC2::Instance'  
    Properties:  
      ImageId: !Ref LatestAmiId  
      InstanceType: t2.micro  
      SecurityGroupIds:  
        - !FindInMap [SecurityGroupMap, !Ref "AWS::Region", !Ref EnvironmentType]
```

Sumber daya terkait

Topik terkait ini dapat membantu saat Anda mengembangkan templat yang menggunakan `Fn::FindInMap` fungsi tersebut.

- [Fn::FindInMap](#)
- [Fn::FindInMapperangkat tambahan](#)

- [Fn::Sub](#)

CloudFormation Metadatasintaks template

Metadatatamenyimpan informasi tambahan menggunakan objek JSON atau YAMAL. Jenis metadata tingkat template yang dapat Anda gunakan dalam template Anda meliputi:

Metadata kustom

Menyimpan pasangan nilai kunci yang ditentukan pengguna. Misalnya, Anda dapat memberikan informasi tambahan yang tidak memengaruhi pembuatan sumber daya tetapi menawarkan konteks tambahan tentang spesifikasi infrastruktur, tim, atau penerapan.

`AWS::CloudFormation::Interface`

Mendefinisikan pengelompokan dan urutan parameter input saat ditampilkan di konsol. CloudFormation Secara default, CloudFormation konsol mengurutkan parameter menurut abjad berdasarkan ID logisnya.

`AWS::CloudFormation::Designer`

AWS CloudFormation Desainer (Desainer) mencapai akhir hayat pada 5 Februari 2025.

Important

Selama pembaruan tumpukan, Anda tidak dapat memperbarui bagian Metadata dengan sendirinya. Anda hanya dapat memperbaruinya jika menyertakan perubahan yang menambahkan, mengubah, atau menghapus sumber daya.

CloudFormation tidak mengubah, memodifikasi, atau menyunting informasi apa pun yang Anda sertakan di Metadata bagian ini. Karena itu, kami sangat merekomendasikan agar Anda tidak menggunakan bagian ini untuk menyimpan informasi sensitif, seperti kata sandi atau rahasia.

Sintaksis

Untuk mendeklarasikan metadata kustom di CloudFormation template Anda, gunakan sintaks berikut:

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"
```

Untuk sintaks untuk `AWS::CloudFormation::Interface`, lihat [Mengatur CloudFormation parameter dengan AWS::CloudFormation::Interface metadata](#).

Mengatur CloudFormation parameter dengan **AWS::CloudFormation::Interface** metadata

`AWS::CloudFormation::Interface` adalah kunci metadata yang mendefinisikan bagaimana parameter dikelompokkan dan diurutkan di konsol. CloudFormation Secara default, konsol mencantumkan parameter input dalam urutan abjad menurut logisnya IDs saat Anda membuat atau memperbarui tumpukan di konsol. Dengan menggunakan kunci ini, Anda dapat menentukan pengelompokan parameter dan pengurutan Anda sendiri agar pengguna dapat secara efisien menentukan nilai parameter. Misalnya, Anda dapat mengelompokkan semua parameter EC2 terkait dalam satu grup dan semua parameter terkait VPC di grup lain.

Pada kunci metadata, Anda dapat menentukan grup yang akan dibuat, parameter yang akan disertakan di setiap grup, dan urutan saat konsol menampilkan setiap parameter dalam kelompoknya.

Anda juga dapat menentukan label untuk parameter. Label adalah nama atau deskripsi yang mudah dimengerti yang ditampilkan oleh konsol sebagai pengganti ID logis parameter. Label berguna untuk membantu pengguna memahami nilai yang ditentukan untuk setiap parameter. Misalnya, Anda dapat melabeli parameter `KeyPair Select` an EC2 key pair.

Semua parameter yang Anda referensikan dalam kunci metadata harus dinyatakan dalam `Parameters` bagian dari template.

 Note

Hanya CloudFormation konsol yang menggunakan kunci `AWS::CloudFormation::Interface` metadata. AWS CLI dan panggilan API tidak menggunakan kunci ini.

Sintaksis

Untuk mendeklarasikan entitas ini di CloudFormation template Anda, gunakan sintaks berikut:

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
```

Properti

ParameterGroups

Daftar jenis grup parameter, di mana Anda menentukan nama grup, parameter di setiap grup, dan urutan parameter yang ditampilkan.

Wajib: Tidak

Label

Nama untuk grup parameter.

Wajib: Tidak

default

Label default yang digunakan CloudFormation konsol untuk memberi nama grup parameter.

Wajib: Tidak

Tipe: String

Parameters

Daftar parameter case-sensitive logis IDs untuk disertakan dalam grup. Parameter harus sudah didefinisikan dalam `Parameters` bagian dari templat. Parameter dapat dimasukkan hanya dalam satu kelompok parameter.

Konsol mencantumkan parameter yang tidak Anda kaitkan dengan grup parameter dalam urutan abjad di `Other parameters` grup.

Wajib: Tidak

Jenis: Daftar nilai String

ParameterLabels

Pemetaan parameter dan nama ramahnya yang ditampilkan CloudFormation konsol saat tumpukan dibuat atau diperbarui.

Wajib: Tidak

Label parameter

Label untuk parameter. Label mendefinisikan nama atau deskripsi ramah yang ditampilkan CloudFormation konsol di halaman Tentukan Parameter saat tumpukan dibuat atau diperbarui. Label parameter harus berupa ID logis peka huruf besar/kecil dari parameter yang valid yang telah dideklarasikan di `Parameters` bagian template.

Wajib: Tidak

`default`

Label default yang digunakan CloudFormation konsol untuk memberi nama parameter.

Wajib: Tidak

Tipe: String

Contoh

Contoh berikut menentukan dua grup parameter: `Network Configuration` dan `Amazon EC2 Configuration`. Grup `Network Configuration` mencakup parameter `VPCID`, `SubnetId`, dan `SecurityGroupId`, yang ditentukan di bagian `Parameters` dari templat (tidak ditampilkan). Urutan di mana konsol menunjukkan parameter ini ditentukan berdasarkan urutan parameter yang tercantum, yang dimulai dengan parameter `VPCID`. Contoh secara identik mengelompokkan dan mengurutkan parameter `Amazon EC2 Configuration`.

Contoh ini juga menentukan label untuk parameter `VPCID`. Konsol akan menunjukkan Di VPC mana ini harus di-deploy? dan bukan ID logis parameter (`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?"
```

Grup parameter di konsol

Penggunaan kunci metadata dari contoh ini, gambar berikut menunjukkan cara konsol menampilkan grup parameter saat tumpukan dibuat atau diperbarui: Grup parameter di konsol

CloudFormation Rules sintaks template

RulesBagian ini adalah bagian opsional dari CloudFormation template yang memungkinkan logika validasi kustom. Saat disertakan, bagian ini berisi fungsi aturan yang memvalidasi nilai parameter sebelum CloudFormation membuat atau memperbarui sumber daya apa pun.

Aturan berguna ketika batasan parameter standar tidak mencukupi. Misalnya, ketika SSL diaktifkan, sertifikat dan nama domain harus disediakan. Aturan dapat memastikan bahwa dependensi ini terpenuhi.

Sintaksis

RulesBagian ini menggunakan sintaks berikut:

JSON

Bagian Rules dari templat terdiri dari nama kunci Rules, diikuti oleh titik dua. Anda harus menggunakan tanda kurung untuk membungkus seluruh deklarasi aturan. Jika Anda mendeklarasikan beberapa aturan, mereka dibatasi oleh koma. Untuk setiap aturan, Anda mendeklarasi nama logis dalam tanda kutip diikuti oleh titik dua dan tanda kurung yang membungkus syarat dan pernyataan aturan.

```
{
```

```

"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

```

Bidang aturan

Bagian Rules dapat mencakup bidang-bidang berikut.

Logical ID (juga disebut nama logis)

Pengidentifikasi unik untuk setiap aturan.

RuleCondition (opsional)

Properti yang menentukan kapan aturan berlaku. Jika Anda tidak menentukan syarat aturan, pernyataan aturan selalu berlaku. Untuk setiap aturan, Anda hanya dapat menentukan satu syarat aturan.

Assertions(diperlukan)

Satu atau lebih pernyataan yang menentukan nilai yang dapat diterima untuk parameter tertentu.

Assert

Suatu kondisi yang harus dievaluasi `true`.

AssertDescription

Pesan yang ditampilkan saat pernyataan gagal.

Fungsi intrinsik khusus aturan

Untuk menentukan aturan Anda, Anda harus menggunakan fungsi khusus aturan, yang merupakan fungsi yang hanya dapat digunakan di Rules bagian templat. Sementara fungsi-fungsi ini dapat bersarang, hasil akhir dari kondisi aturan atau pernyataan harus salah satu atau `true` `false`

Fungsi aturan berikut tersedia:

- [Fn::And](#)
- [Fn::Contains](#)

- [Fn::EachMemberEquals](#)
- [Fn::EachMemberIn](#)
- [Fn::Equals](#)
- [Fn::If](#)
- [Fn::Not](#)
- [Fn::Or](#)
- [Fn::RefAll](#)
- [Fn::ValueOf](#)
- [Fn::ValueOfAll](#)

Fungsi-fungsi ini digunakan dalam kondisi atau pernyataan aturan. Properti kondisi menentukan apakah CloudFormation berlaku pernyataan. Jika kondisi mengevaluasi `true`, CloudFormation mengevaluasi pernyataan untuk memverifikasi apakah nilai parameter valid ketika produk yang disediakan dibuat atau diperbarui. Jika nilai parameter tidak valid, CloudFormation tidak membuat atau memperbarui tumpukan. Jika kondisi dievaluasi `false`, CloudFormation tidak memeriksa nilai parameter dan melanjutkan operasi tumpukan.

Contoh

Topik

- [Memverifikasi nilai parameter secara bersyarat](#)
- [Validasi lintas parameter](#)

Memverifikasi nilai parameter secara bersyarat

Dalam contoh berikut, dua aturan tersebut memeriksa nilai parameter `InstanceType`. Tergantung pada nilai parameter lingkungan (`test` atau `prod`), pengguna harus menentukan `a1.medium` atau `a1.large` untuk parameter `InstanceType`. Parameter `InstanceType` dan `Environment` harus dideklarasikan dalam bagian `Parameters` dari templat yang sama.

JSON

```
{
  "Rules": {
    "testInstanceType": {
      "RuleCondition": {
```

```

        "Fn::Equals": [
            {"Ref": "Environment"},
            "test"
        ]
    },
    "Assertions": [
        {
            "Assert": {
                "Fn::Contains": [
                    ["a1.medium"],
                    {"Ref": "InstanceType"}
                ]
            },
            "AssertDescription": "For a test environment, the instance type must be
a1.medium"
        }
    ],
    "prodInstanceType": {
        "RuleCondition": {
            "Fn::Equals": [
                {"Ref": "Environment"},
                "prod"
            ]
        },
        "Assertions": [
            {
                "Assert": {
                    "Fn::Contains": [
                        ["a1.large"],
                        {"Ref": "InstanceType"}
                    ]
                },
                "AssertDescription": "For a production environment, the instance type must be
a1.large"
            }
        ]
    }
}

```

YAML

```
Rules:
  testInstanceType:
    RuleCondition: !Equals
      - !Ref Environment
      - test
    Assertions:
      - Assert:
          'Fn::Contains':
            - a1.medium
            - !Ref InstanceType
          AssertDescription: 'For a test environment, the instance type must be a1.medium'
      - prodInstanceType:
          RuleCondition: !Equals
            - !Ref Environment
            - prod
          Assertions:
            - Assert:
                'Fn::Contains':
                  - a1.large
                  - !Ref InstanceType
                AssertDescription: 'For a production environment, the instance type must be a1.large'
```

Validasi lintas parameter

Contoh template berikut menunjukkan penggunaan aturan untuk validasi lintas parameter. Mereka membuat situs web sampel yang berjalan pada grup Auto Scaling di belakang penyeimbang beban. Situs web tersedia di port 80 atau 443 tergantung pada parameter input. Instans dalam grup Auto Scaling dapat dikonfigurasi untuk mendengarkan pada port apa pun (dengan 8888 sebagai default).

Aturan dalam template ini memvalidasi parameter input sebelum pembuatan stack. Mereka memverifikasi bahwa semua subnet milik VPC yang ditentukan dan memastikan bahwa ketika UseSSL parameter disetel Yes ke, baik sertifikat SSL ARN dan nama zona yang dihosting disediakan.

 Note

Anda akan ditagih untuk AWS sumber daya yang digunakan jika Anda membuat tumpukan dari template ini.

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"
    }
  ]
}
},
"Mappings": {
  "AWSAMIRegionMap": {
    "us-east-1": {"AMZNLinuxHVM": "ami-0ff8a91507f77f867"},
    "us-west-1": {"AMZNLinuxHVM": "ami-0bdb828fd58c52235"},
    "eu-west-1": {"AMZNLinuxHVM": "ami-047bb4163c506cd98"},
    "ap-southeast-1": {"AMZNLinuxHVM": "ami-08569b978cc4dfa10"}
  }
},
"Resources": {
  "WebServerGroup": {
    "Type": "AWS::AutoScaling::AutoScalingGroup",
    "Properties": {
      "VPCZoneIdentifier": {"Ref": "Subnets"},
      "LaunchConfigurationName": {"Ref": "LaunchConfig"},
      "MinSize": "2",
      "MaxSize": "2",
      "TargetGroupARNs": [{"Ref": "ALBTargetGroup"}]
    },
    "CreationPolicy": {
      "ResourceSignal": {"Timeout": "PT15M"}
    },
    "UpdatePolicy": {
      "AutoScalingRollingUpdate": {
        "MinInstancesInService": "1",
        "MaxBatchSize": "1",
        "PauseTime": "PT15M",
        "WaitOnResourceSignals": true
      }
    }
  },
  "LaunchConfig": {
    "Type": "AWS::AutoScaling::LaunchConfiguration",
    "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.LaunchConfig.Metadata.AWS::CloudFormation::Init\n",
        "action=/opt/aws/bin/cfn-init -v ",
        "      --stack ", {"Ref": "AWS::StackName"},
        "      --resource LaunchConfig ",
        "      --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": {
        "ImageId": {"Fn::FindInMap": ["AWSAMIRegionMap", {"Ref": "AWS::Region"},
"AMZNLinuxHVM"]},
        "SecurityGroups": [{"Ref": "InstanceSecurityGroup"}],
        "InstanceType": {"Ref": "InstanceType"},
        "KeyName": {"Ref": "KeyName"},
        "UserData": {
            "Fn::Base64": {"Fn::Join": ["", [
                "#!/bin/bash -xe\n",
                "yum update -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"
            ]]}
        }
    },
    "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
Mappings:
  AWSAMIRegionMap:
    us-east-1:
      AMZNINUXHVM: ami-0ff8a91507f77f867
    us-west-1:
      AMZNINUXHVM: ami-0bdb828fd58c52235
    eu-west-1:
      AMZNINUXHVM: ami-047bb4163c506cd98
    ap-southeast-1:
      AMZNINUXHVM: ami-08569b978cc4dfa10

```



```

Resources:
  WebServerGroup:
    Type: AWS::AutoScaling::AutoScalingGroup
    Properties:
      VPCZoneIdentifier: !Ref Subnets
      LaunchConfigurationName: !Ref LaunchConfig
      MinSize: '2'
      MaxSize: '2'
      TargetGroupARNs:
        - !Ref ALBTargetGroup
    CreationPolicy:
      ResourceSignal:
        Timeout: PT15M
    UpdatePolicy:
      AutoScalingRollingUpdate:
        MinInstancesInService: '1'
        MaxBatchSize: '1'
        PauseTime: PT15M
        WaitOnResourceSignals: 'true'
  LaunchConfig:
    Type: AWS::AutoScaling::LaunchConfiguration
    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.LaunchConfig.Metadata.AWS::CloudFormation::Init
            action=/opt/aws/bin/cfn-init -v --stack ${AWS::StackName} --resource
LaunchConfig --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:
        ImageId: !FindInMap
            - AWSAMIRegionMap
            - !Ref 'AWS::Region'
            - AMZNLinuxHVM
        SecurityGroups:
            - !Ref InstanceSecurityGroup
        InstanceType: !Ref InstanceType
        KeyName: !Ref KeyName
        UserData: !Base64
            Fn::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}
        ELBSecurityGroup:
            Type: AWS::EC2::SecurityGroup

```

Properties:

GroupDescription: Allow access to the ELB

VpcId: !Ref VpcId

SecurityGroupIngress:

- !If
 - UseALBSSL
 - IpProtocol: tcp
 - FromPort: 443
 - ToPort: 443
 - CidrIp: 0.0.0.0/0
- IpProtocol: tcp
- FromPort: 80
- ToPort: 80
- CidrIp: 0.0.0.0/0

ApplicationLoadBalancer:

Type: AWS::ElasticLoadBalancingV2::LoadBalancer

Properties:

Subnets: !Ref Subnets

SecurityGroups:

- !Ref ELBSecurityGroup

ALBListener:

Type: AWS::ElasticLoadBalancingV2::Listener

Properties:

DefaultActions:

- Type: forward
 - TargetGroupArn: !Ref ALBTargetGroup
- LoadBalancerArn: !Ref ApplicationLoadBalancer

Port: !If

- UseALBSSL
- 443
- 80

Protocol: !If

- UseALBSSL
- HTTPS
- HTTP

Certificates:

- !If
 - UseALBSSL
 - CertificateArn: !Ref ALBSSLCertificateARN
- !Ref 'AWS::NoValue'

ALBTargetGroup:

Type: AWS::ElasticLoadBalancingV2::TargetGroup

Properties:

HealthCheckIntervalSeconds: 30

```

    HealthCheckTimeoutSeconds: 5
    HealthyThresholdCount: 3
    Port: 80
    Protocol: HTTP
    UnhealthyThresholdCount: 5
    VpcId: !Ref VpcId
InstanceSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Enable SSH access and HTTP access on the inbound port
    SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: 80
        ToPort: 80
        SourceSecurityGroupId: !Select
          - 0
          - !GetAtt
            - ApplicationLoadBalancer
            - SecurityGroups
      - IpProtocol: tcp
        FromPort: 22
        ToPort: 22
        CidrIp: !Ref SSHLocation
    VpcId: !Ref VpcId
RecordSet:
  Type: AWS::Route53::RecordSetGroup
  Condition: UseALBSSL
  Properties:
    HostedZoneName: !Join
      - ''
      - - !Ref HostedZoneName
        - .
    RecordSets:
      - Name: !Join
          - ''
          - - !Select
              - '0'
              - !Split
                - .
                - !GetAtt
                  - ApplicationLoadBalancer
                  - DNSName
            - .
            - !Ref HostedZoneName

```

```

    - .
    Type: A
    AliasTarget:
      DNSName: !GetAtt
        - ApplicationLoadBalancer
        - DNSName
      EvaluateTargetHealth: true
      HostedZoneId: !GetAtt
        - ApplicationLoadBalancer
        - CanonicalHostedZoneID
  Outputs:
    URL:
      Description: URL of the website
      Value: !Join
        - ''
        - - !If
            - UseALBSSL
            - !Join
              - ''
              - - 'https://'
              - !Join
                - ''
                - - !Select
                    - '0'
                    - !Split
                      - .
                      - !GetAtt
                        - ApplicationLoadBalancer
                        - DNSName
                - .
                - !Ref HostedZoneName
                - .
            - !Join
              - ''
              - - 'http://'
              - !GetAtt
                - ApplicationLoadBalancer
                - DNSName

```

CloudFormation Conditionssintaks template

Bagian Conditions opsional berisi pernyataan yang menentukan keadaan di mana entitas dibuat atau dikonfigurasi. Misalnya, Anda dapat membuat kondisi dan mengaitkannya dengan sumber daya

atau output sehingga CloudFormation menciptakan sumber daya atau output hanya jika kondisi benar. Demikian pula, Anda dapat mengaitkan kondisi dengan properti sehingga CloudFormation menetapkan properti ke nilai tertentu hanya jika kondisi benar. Jika kondisinya salah, CloudFormation tetapkan properti ke nilai alternatif yang Anda tentukan.

Anda dapat menggunakan kondisi saat ingin menggunakan kembali template untuk membuat sumber daya dalam konteks yang berbeda, seperti lingkungan pengujian versus produksi. Misalnya, dalam template Anda, Anda dapat menambahkan parameter `EnvironmentType` input yang menerima salah satu `prod` atau `test` sebagai input. Untuk `prod` lingkungan, Anda mungkin menyertakan EC2 contoh dengan kemampuan tertentu, sedangkan untuk `test` lingkungan, Anda mungkin menggunakan kemampuan yang lebih rendah untuk menghemat uang. Definisi kondisi ini memungkinkan Anda untuk menentukan sumber daya mana yang dibuat dan bagaimana mereka dikonfigurasi untuk setiap jenis lingkungan.

Sintaksis

Bagian `Conditions` terdiri dari nama kunci `Conditions`. Setiap deklarasi kondisi mencakup ID logis dan satu atau lebih fungsi intrinsik.

JSON

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

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

YAML

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

  LogicalConditionName2:
    Intrinsic function:
      ...
```

Bagaimana kondisi bekerja

Untuk menggunakan kondisi, ikuti langkah-langkah ini:

1. Tambahkan definisi parameter - Tentukan input yang akan dievaluasi kondisi Anda di `Parameters` bagian template Anda. Kondisi mengevaluasi benar atau salah berdasarkan nilai parameter masukan ini. Perhatikan bahwa parameter semu tersedia secara otomatis dan tidak memerlukan definisi eksplisit di bagian tersebut. `Parameters` Untuk informasi selengkapnya tentang parameter semu, lihat [Dapatkan AWS nilai menggunakan parameter semu](#).
2. Tambahkan definisi kondisi - Tentukan kondisi di `Conditions` bagian menggunakan fungsi intrinsik seperti `Fn::If`, `Fn::Equals`, dan `Fn::Not`. Kondisi ini menentukan kapan CloudFormation menciptakan sumber daya terkait. Kondisi dapat didasarkan pada:
 - Nilai parameter masukan atau semu
 - Syarat lainnya
 - Nilai pemetaan
3. Mengaitkan kondisi dengan sumber daya atau output — Kondisi referensi dalam sumber daya atau output menggunakan `Condition` kunci dan ID logis kondisi. Secara opsional, gunakan `Fn::If` di bagian lain dari template (seperti nilai properti) untuk menetapkan nilai berdasarkan kondisi.

CloudFormation mengevaluasi kondisi saat membuat atau memperbarui tumpukan. CloudFormation menciptakan entitas yang terkait dengan kondisi sebenarnya dan mengabaikan entitas yang terkait dengan kondisi palsu. CloudFormation juga mengevaluasi kembali kondisi ini selama setiap pembaruan tumpukan sebelum memodifikasi sumber daya apa pun. Entitas yang tetap terkait dengan kondisi sebenarnya diperbarui, sedangkan entitas yang terkait dengan kondisi palsu dihapus.

Important

Selama pembaruan tumpukan, Anda tidak dapat memperbarui syarat dengan sendirinya. Anda hanya dapat memperbarui syarat jika menyertakan perubahan yang menambahkan, mengubah, atau menghapus sumber daya.

Fungsi intrinsik syarat

Anda dapat menggunakan fungsi intrinsik berikut untuk menentukan syarat:

- [Fn::And](#)

- [Fn::Equals](#)
- [Fn::ForEach](#)
- [Fn::If](#)
- [Fn::Not](#)
- [Fn::Or](#)

Note

Fn::If hanya didukung dalam atribut metadata, atribut kebijakan pembaruan, dan nilai properti di bagian Resources dan bagian Outputs dari templat.

Contoh

Topik

- [Pembuatan sumber daya berbasis lingkungan](#)
- [Penyediaan sumber daya multi-kondisi](#)

Pembuatan sumber daya berbasis lingkungan

Contoh berikut ini menyediakan EC2 instance, dan secara kondisional membuat dan melampirkan volume EBS baru hanya jika tipe lingkungannya. prod Jika lingkungannya test, mereka hanya membuat EC2 instance tanpa volume tambahan.

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Parameters": {
    "EnvType": {
      "Description": "Environment type",
      "Default": "test",
      "Type": "String",
      "AllowedValues": [
        "prod",
        "test"
      ],
      "ConstraintDescription": "must specify prod or test"
    }
  }
```



```

    }
  },
  "Conditions": {
    "CreateProdResources": {
      "Fn::Equals": [
        {
          "Ref": "EnvType"
        },
        "prod"
      ]
    }
  },
  "Resources": {
    "EC2Instance": {
      "Type": "AWS::EC2::Instance",
      "Properties": {
        "ImageId": "ami-1234567890abcdef0"
      }
    },
    "MountPoint": {
      "Type": "AWS::EC2::VolumeAttachment",
      "Condition": "CreateProdResources",
      "Properties": {
        "InstanceId": {
          "Ref": "EC2Instance"
        },
        "VolumeId": {
          "Ref": "NewVolume"
        },
        "Device": "/dev/sdh"
      }
    },
    "NewVolume": {
      "Type": "AWS::EC2::Volume",
      "Condition": "CreateProdResources",
      "Properties": {
        "Size": 100,
        "AvailabilityZone": {
          "Fn::GetAtt": [
            "EC2Instance",
            "AvailabilityZone"
          ]
        }
      }
    }
  }
}

```

```
    }  
  }  
}
```

YAML

```
AWS::TemplateFormatVersion: 2010-09-09  
Parameters:  
  EnvType:  
    Description: Environment type  
    Default: test  
    Type: String  
    AllowedValues:  
      - prod  
      - test  
    ConstraintDescription: must specify prod or test  
Conditions:  
  CreateProdResources: !Equals  
    - !Ref EnvType  
    - prod  
Resources:  
  EC2Instance:  
    Type: AWS::EC2::Instance  
    Properties:  
      ImageId: ami-1234567890abcdef0  
  MountPoint:  
    Type: AWS::EC2::VolumeAttachment  
    Condition: CreateProdResources  
    Properties:  
      InstanceId: !Ref EC2Instance  
      VolumeId: !Ref NewVolume  
      Device: /dev/sdh  
  NewVolume:  
    Type: AWS::EC2::Volume  
    Condition: CreateProdResources  
    Properties:  
      Size: 100  
      AvailabilityZone: !GetAtt  
        - EC2Instance  
        - AvailabilityZone
```

Penyediaan sumber daya multi-kondisi

Contoh berikut membuat bucket S3 secara kondisional jika nama bucket disediakan, dan lampirkan kebijakan bucket hanya jika lingkungan disetel ke. prod Jika tidak ada nama bucket yang diberikan atau lingkungannyatest, tidak ada sumber daya yang dibuat.

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: ...

```

CloudFormation Transform bagian template

Transform bagian opsional menentukan satu atau lebih makro yang CloudFormation digunakan untuk memproses template Anda dengan cara tertentu.

Makro dapat melakukan tugas-tugas sederhana seperti menemukan dan mengganti teks, atau mereka dapat membuat transformasi yang lebih luas ke seluruh template. CloudFormation mengeksekusi makro dalam urutan yang ditentukan. Saat Anda membuat set perubahan, CloudFormation buat set perubahan yang menyertakan konten template yang diproses. Anda kemudian dapat meninjau perubahan dan mengeksekusi set perubahan. Untuk informasi selengkapnya tentang cara kerja makro, lihat [Lakukan pemrosesan kustom pada CloudFormation template dengan makro template](#).

CloudFormation juga mendukung transformasi, yang merupakan makro yang dihosting oleh CloudFormation. CloudFormation memperlakukan transformasi ini sama seperti makro apa pun yang Anda buat dalam hal urutan eksekusi dan cakupan. Untuk informasi selengkapnya, lihat [Referensi Transform](#).

Untuk mendeklarasikan beberapa makro, gunakan format daftar dan tentukan satu atau lebih makro.

Misalnya, dalam contoh templat di bawah ini `AWS::Serverless`, CloudFormation mengevaluasi `MyMacro` dan kemudian, keduanya dapat memproses isi seluruh templat karena dimasukkan ke dalam `Transform` bagian.

```
# 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 format template sintaks versi

`AWSTemplateFormatVersion` bagian (opsional) mengidentifikasi versi format template yang sesuai dengan template. Versi format templat terbaru adalah `2010-09-09` dan saat ini merupakan satu-satunya nilai yang valid.

Versi format template tidak sama dengan versi API. Versi format template dapat berubah secara independen dari versi API.

Nilai untuk deklarasi versi format templat harus berupa literal string. Anda tidak dapat menggunakan parameter atau fungsi untuk menentukan versi format templat. Jika Anda tidak menentukan nilai,

CloudFormation asumsikan versi format template terbaru. Cuplikan berikut ini adalah contoh deklarasi versi format templat yang valid:

JSON

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

YAML

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

CloudFormation template Description sintaksis

DescriptionBagian (opsional) memungkinkan Anda untuk menyertakan string teks yang menggambarkan template. Bagian ini harus selalu mengikuti bagian versi format templat.

Nilai untuk deklarasi deskripsi harus berupa string literal yang panjangnya antara 0 dan 1024 byte. Anda tidak dapat menggunakan parameter atau fungsi untuk menentukan deskripsi. Cuplikan berikut ini adalah contoh deklarasi deskripsi:

Important

Selama pembaruan tumpukan, Anda tidak dapat memperbarui bagian Description dengan sendirinya. Anda hanya dapat memperbaruinya jika menyertakan perubahan yang menambahkan, mengubah, atau menghapus sumber daya.

JSON

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

YAML

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

Buat template secara visual dengan Infrastructure Composer

AWS Infrastructure Composer (sebelumnya dikenal sebagai Application Composer) membantu Anda menyusun dan mengkonfigurasi aplikasi modern secara visual. AWS Alih-alih menulis kode, Anda dapat menarik dan melepas sumber daya yang berbeda untuk membangun aplikasi Anda secara visual.

Infrastructure Composer dalam mode CloudFormation konsol adalah alat yang disarankan untuk bekerja dengan CloudFormation template secara visual. Versi Infrastructure Composer yang dapat Anda akses dari CloudFormation konsol ini merupakan peningkatan dari alat lama yang disebut AWS CloudFormation Designer.

Dengan Infrastructure Composer dalam mode CloudFormation konsol, Anda dapat menyeret, melepas, mengonfigurasi, dan menghubungkan berbagai sumber daya, yang disebut kartu, ke kanvas visual. Pendekatan visual ini memudahkan untuk merancang dan mengedit arsitektur aplikasi Anda tanpa harus bekerja dengan template secara langsung. Untuk mengakses mode ini dari [CloudFormation konsol](#), pilih Infrastructure Composer dari menu navigasi sisi kiri.

Untuk informasi selengkapnya, lihat [Cara menulis AWS Infrastructure Composer di Panduan AWS Infrastructure Composer Pengembang](#).

Mengapa menggunakan Infrastructure Composer dalam mode CloudFormation konsol?

Memvisualisasikan template Anda di Infrastructure Composer membantu Anda mengidentifikasi kesenjangan dan area peningkatan dalam CloudFormation templat dan arsitektur aplikasi Anda. Infrastructure Composer meningkatkan pengalaman pengembangan Anda dengan kemudahan dan efisiensi membangun dan memodifikasi CloudFormation tumpukan secara visual. Anda dapat memulai dengan draf awal, membuat kode deployable, dan menggabungkan alur kerja pengembang Anda dengan desainer visual di Infrastructure Composer.

Bagaimana mode ini berbeda dari konsol Infrastructure Composer?

Sementara versi CloudFormation konsol Infrastructure Composer memiliki fitur yang mirip dengan konsol Infrastructure Composer standar, ada beberapa perbedaan. Kartu terkait Lambda (Fungsi Lambda dan Lapisan Lambda) memerlukan pembuatan kode dan solusi pengemasan yang tidak tersedia di Infrastructure Composer dalam mode konsol. CloudFormation Sinkronisasi lokal juga tidak tersedia dalam mode ini.

Namun, Anda dapat menggunakan kartu terkait Lambda ini dan fitur sinkronisasi lokal di [konsol Infrastructure Composer](#) atau [AWS Toolkit for Visual Studio Code](#) Untuk informasi selengkapnya, lihat [Panduan AWS Infrastructure Composer Pengembang](#) dan [Komposer Infrastruktur](#) di Panduan [AWS Toolkit for Visual Studio Code Pengguna](#).

Hasilkan templat dari sumber daya yang ada dengan generator IAc

Dengan CloudFormation infrastruktur sebagai generator kode (IAc generator), Anda dapat membuat template menggunakan AWS sumber daya yang disediakan di akun Anda yang belum dikelola oleh CloudFormation

Berikut ini adalah manfaat dari generator IAc:

- Bawa seluruh aplikasi di bawah CloudFormation manajemen atau migrasikan ke dalam AWS CDK aplikasi.
- Hasilkan template tanpa harus mendeskripsikan properti sumber daya berdasarkan properti dan kemudian menerjemahkannya ke dalam sintaks JSON atau YAMAL.
- Gunakan templat untuk mereplikasi sumber daya di akun atau Wilayah baru.

Proses pembuatan IAc terdiri dari langkah-langkah berikut:

1. Pindai sumber daya — Langkah pertama adalah memulai pemindaian sumber daya Anda. Pemindaian ini di seluruh wilayah dan kedaluwarsa setelah 30 hari. Selama waktu ini, Anda dapat membuat beberapa templat dari pemindaian yang sama.
2. Buat template Anda - Untuk membuat template, Anda memiliki dua opsi:
 - Buat template baru dari awal dan tambahkan sumber daya yang dipindai dan sumber daya terkait ke dalamnya.
 - Gunakan CloudFormation tumpukan yang ada sebagai titik awal dan tambahkan sumber daya yang dipindai dan sumber daya terkait ke templatnya.
3. Impor sumber daya — Gunakan templat Anda untuk mengimpor sumber daya sebagai CloudFormation tumpukan atau memigrasikannya ke dalam AWS CDK aplikasi.

Fitur generator IAC tersedia di semua Wilayah komersial dan mendukung banyak jenis AWS sumber daya umum. Untuk daftar lengkap sumber daya yang didukung, lihat [Dukungan jenis sumber daya](#).

Topik

- [Pertimbangan](#)
- [Izin IAM diperlukan untuk memindai sumber daya](#)
- [Perintah yang umum digunakan untuk pembuatan template, manajemen, dan penghapusan](#)
- [Migrasikan template ke AWS CDK](#)
- [Mulai pemindaian sumber daya dengan CloudFormation generator IAC](#)
- [Lihat ringkasan pemindaian di CloudFormation konsol](#)
- [Buat CloudFormation template dari sumber daya yang dipindai dengan generator IAC](#)
- [Buat CloudFormation tumpukan dari sumber daya yang dipindai](#)
- [Selesaikan properti hanya tulis](#)

Pertimbangan

Anda dapat membuat template JSON atau YAMAL untuk AWS sumber daya yang telah Anda baca. Template untuk kemampuan generator IAC memodelkan sumber daya cloud dengan andal dan cepat tanpa harus mendeskripsikan properti sumber daya berdasarkan properti.

Tabel berikut mencantumkan kuota yang tersedia untuk fitur generasi IAC.

Nama	Pemindaian penuh	Pemindaian sebagian
Jumlah maksimum sumber daya yang dapat diproses dalam pemindaian	100.000	100.000
Jumlah pemindaian per hari (untuk pemindaian dengan kurang dari 10.000 sumber daya)	10	10
Jumlah pemindaian per hari (untuk pemindaian dengan lebih dari 10.000 sumber daya)	1	1
Jumlah template yang dihasilkan secara bersamaan per akun	5	5
Jumlah sumber daya bersamaan yang dimodelkan untuk satu pembuatan templat	5	5

Nama	Pemindaian penuh	Pemindaian sebagian
Jumlah total sumber daya yang dapat dimodelkan dalam satu template	500	500
Jumlah maksimum template yang dihasilkan per akun	1.000	1.000

Important

Generator IAC hanya mendukung AWS sumber daya yang didukung oleh Cloud Control API di Wilayah Anda. Untuk informasi selengkapnya, lihat [Dukungan jenis sumber daya](#).

Izin IAM diperlukan untuk memindai sumber daya

Untuk memindai sumber daya dengan generator IAC, prinsipal IAM Anda (pengguna, peran, atau grup) harus memiliki:

- CloudFormation izin pemindaian
- Baca izin untuk layanan target AWS

Cakupan pemindaian terbatas pada sumber daya yang telah Anda baca akses. Izin yang hilang tidak akan menyebabkan kegagalan pemindaian tetapi akan mengecualikan sumber daya tersebut.

Untuk contoh kebijakan IAM yang memberikan izin pemindaian dan pengelolaan templat, lihat [Izinkan semua operasi generator IAC](#)

Perintah yang umum digunakan untuk pembuatan template, manajemen, dan penghapusan

Perintah yang umum digunakan untuk bekerja dengan generator IAC meliputi:

- [start-resource-scan](#) untuk memulai pemindaian sumber daya di akun di file Wilayah AWS.
- [describe-resource-scan](#) untuk memantau kemajuan pemindaian sumber daya.
- [list-resource-scans](#) untuk membuat daftar pemindaian sumber daya di file Wilayah AWS.

- [list-resource-scan-resources](#) untuk membuat daftar sumber daya yang ditemukan selama pemindaian sumber daya.
- [list-resource-scan-related-resource](#) untuk membuat daftar sumber daya yang terkait dengan sumber daya yang dipindai.
- [create-generated-template](#) untuk menghasilkan CloudFormation template dari satu set sumber daya yang dipindai.
- [update-generated-template](#) untuk memperbarui template yang dihasilkan.
- [describe-generated-template](#) untuk mengembalikan informasi tentang template yang dihasilkan.
- [list-generated-templates](#) untuk mencantumkan semua templat yang dihasilkan di akun Anda dan Wilayah saat ini.
- [delete-generated-template](#) untuk menghapus template yang dihasilkan.

Migrasikan template ke AWS CDK

AWS Cloud Development Kit (AWS CDK) Ini adalah kerangka pengembangan perangkat lunak open-source yang dapat Anda gunakan untuk mengembangkan, mengelola, dan menyebarkan CloudFormation sumber daya menggunakan bahasa pemrograman populer.

AWS CDK CLI menyediakan integrasi dengan generator IAC. Gunakan `cdk migrate` perintah AWS CDK CLI untuk mengonversi CloudFormation template dan membuat aplikasi CDK baru yang berisi sumber daya Anda. Kemudian, Anda dapat menggunakan AWS CDK untuk mengelola sumber daya Anda dan CloudFormation menerapkannya.

Untuk informasi selengkapnya, lihat [Memigrasi ke AWS CDK](#) dalam Panduan AWS Cloud Development Kit (AWS CDK) Pengembang.

Mulai pemindaian sumber daya dengan CloudFormation generator IAC

Sebelum Anda membuat template dari sumber daya yang ada, Anda harus terlebih dahulu memulai pemindaian sumber daya untuk menemukan sumber daya Anda saat ini dan hubungannya.

Anda dapat memulai pemindaian sumber daya menggunakan salah satu opsi berikut. Untuk pengguna pertama kali generator IAC, kami merekomendasikan opsi pertama.

- Pindai semua sumber daya (pemindaian penuh) - Memindai semua sumber daya yang ada di akun saat ini dan Wilayah. Proses pemindaian ini dapat memakan waktu hingga 10 menit untuk 1.000 sumber daya.

- Pindai sumber daya tertentu (pemindaian sebagian) — Pilih jenis sumber daya mana yang akan dipindai secara manual di akun dan Wilayah saat ini. Opsi ini menyediakan proses pemindaian yang lebih cepat dan lebih terfokus, sehingga ideal untuk pengembangan template berulang.

Setelah pemindaian selesai, Anda dapat memilih sumber daya dan sumber daya terkait yang akan disertakan saat membuat templat Anda. Saat menggunakan pemindaian sebagian, sumber daya terkait hanya akan tersedia selama pembuatan templat jika:

- Anda secara khusus memilihnya sebelum memulai pemindaian, atau
- Mereka diminta untuk menemukan jenis sumber daya yang Anda pilih.

Misalnya, jika Anda memilih `AWS::EKS::Nodegroup` tanpa memilih `AWS::EKS::Cluster`, generator IAC secara otomatis menyertakan `AWS::EKS::Cluster` sumber daya dalam pemindaian karena menemukan grup node memerlukan penemuan cluster terlebih dahulu. Dalam semua kasus lain, pemindaian hanya akan mencakup sumber daya yang Anda pilih secara khusus.

Note

Sebelum melanjutkan, konfirmasikan bahwa Anda memiliki izin yang diperlukan untuk bekerja dengan generator IAC. Untuk informasi selengkapnya, lihat [Izin IAM diperlukan untuk memindai sumber daya](#).

Topik

- [Mulai pemindaian sumber daya \(konsol\)](#)
- [Mulai pemindaian sumber daya \(AWS CLI\)](#)

Mulai pemindaian sumber daya (konsol)

Untuk memulai pemindaian sumber daya dari semua jenis sumber daya (pemindaian penuh)

1. Buka [halaman generator IAC](#) CloudFormation konsol.
2. Pada bilah navigasi di bagian atas layar, pilih Wilayah AWS yang berisi sumber daya untuk dipindai.
3. Dari panel Scan, pilih Mulai pemindaian baru dan kemudian pilih Pindai semua sumber daya.

Untuk memulai pemindaian sumber daya jenis sumber daya tertentu (pemindaian sebagian)

1. Buka [halaman generator IAC](#) CloudFormation konsol.
2. Pada bilah navigasi di bagian atas layar, pilih Wilayah AWS yang berisi sumber daya untuk dipindai.
3. Dari panel Scan, pilih Mulai pemindaian baru dan kemudian pilih Pindai sumber daya tertentu.
4. Dalam kotak dialog Mulai pemindaian sebagian, pilih hingga 100 jenis sumber daya, lalu pilih Mulai pemindaian.

Mulai pemindaian sumber daya (AWS CLI)

Untuk memulai pemindaian sumber daya dari semua jenis sumber daya (pemindaian penuh)

Gunakan perintah berikut [start-resource-scan](#). Ganti *us-east-1* dengan Wilayah AWS yang berisi sumber daya untuk memindai.

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

Jika berhasil, perintah ini mengembalikan ARN pemindaian. Perhatikan ARN di properti ResourceScanId Anda membutuhkannya untuk membuat template Anda.

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

Untuk memulai pemindaian sumber daya jenis sumber daya tertentu (pemindaian sebagian)

1. Gunakan perintah [cat](#) berikut untuk menyimpan jenis sumber daya yang ingin Anda pindai dalam file JSON bernama `config.json` di direktori home Anda. Berikut ini adalah contoh konfigurasi pemindaian yang memindai EC2 instans Amazon, grup keamanan, dan semua sumber daya Amazon S3.

```
$ cat > config.json
[
  {
    "Types": [
      "AWS::EC2::Instance",
```

```

        "AWS::EC2::SecurityGroup",
        "AWS::S3::*"
    ]
}
]

```

- Gunakan [start-resource-scan](#) perintah dengan `--scan-filters` opsi, bersama dengan `config.json` file yang Anda buat, untuk memulai pemindaian sebagian. Ganti `us-east-1` dengan Wilayah AWS yang berisi sumber daya untuk memindai.

```

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

```

Jika berhasil, perintah ini mengembalikan ARN pemindaian. Perhatikan ARN di properti. `ResourceScanId` Anda membutuhkannya untuk membuat template Anda.

```

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

```

Untuk memantau kemajuan pemindaian sumber daya

Gunakan perintah [describe-resource-scan](#). Untuk `--resource-scan-id` opsi, ganti sampel ARN dengan ARN yang sebenarnya.

```

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

```

Jika berhasil, perintah ini mengembalikan output yang serupa dengan yang berikut:

```

{
  "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
}

```

Untuk pemindaian sebagian, output akan terlihat mirip dengan yang berikut:

```

{
  "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::*"  
  ]  
}  
]  
}
```

Untuk deskripsi bidang dalam output, lihat [DescribeResourceScan](#) di Referensi AWS CloudFormation API.

Lihat ringkasan pemindaian di CloudFormation konsol

Setelah pemindaian selesai, Anda dapat melihat visualisasi sumber daya yang ditemukan selama pemindaian untuk membantu Anda mengidentifikasi konsentrasi sumber daya di berbagai jenis produk.

Untuk melihat informasi tentang sumber daya yang ditemukan selama pemindaian

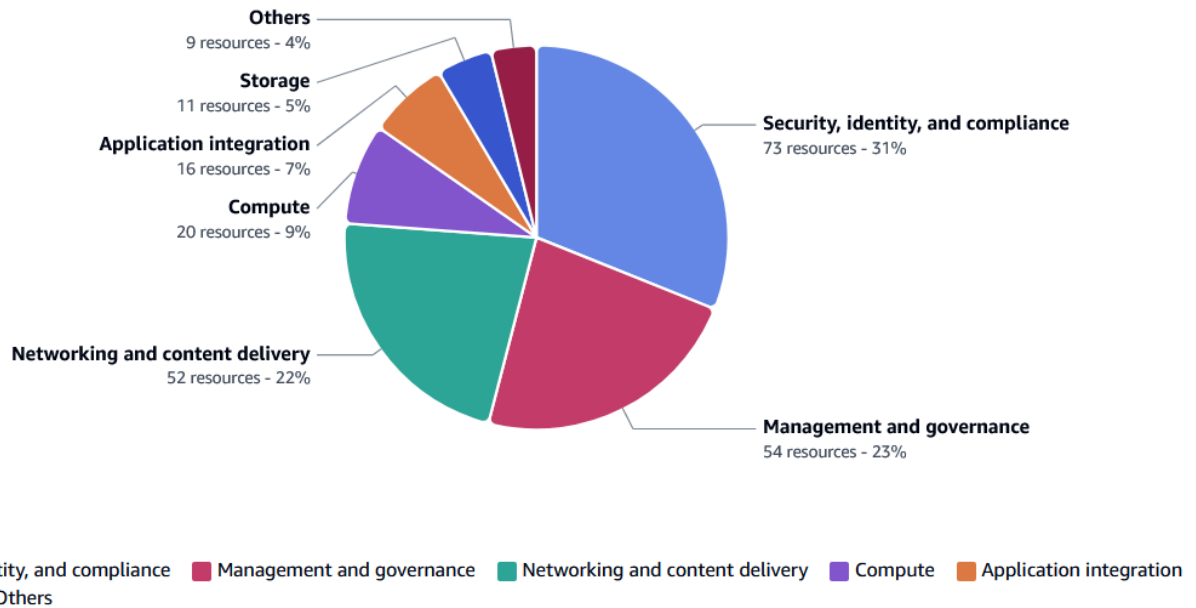
1. Buka [halaman generator IAC](#) CloudFormation konsol.
2. Pada bilah navigasi di bagian atas layar, pilih Wilayah AWS yang berisi pemindaian sumber daya untuk dilihat.
3. Dari panel navigasi, pilih generator IAC.
4. Di bawah rincian sumber daya yang dipindai, Anda akan menemukan rincian visual dari sumber daya yang dipindai berdasarkan jenis produk, misalnya, Compute dan Storage.
5. Untuk menyesuaikan jumlah jenis produk yang ditampilkan, pilih Filter data yang ditampilkan. Ini membantu Anda menyesuaikan visualisasi untuk fokus pada jenis produk yang paling Anda minati.
6. Di sisi kanan halaman adalah panel detail ringkasan Pindai. Untuk membuka panel, pilih ikon panel terbuka.

Scanned resources breakdown 235

View a visual breakdown of your scanned resources by product types. Select a product type for a further breakdown.

Filter displayed data

Filter by product type



Buat CloudFormation template dari sumber daya yang dipindai dengan generator IAC

Topik ini menjelaskan cara membuat template dari sumber daya yang dipindai menggunakan fitur generator IAC.

Buat template dari sumber daya yang dipindai (konsol)

Untuk membuat template tumpukan dari sumber daya yang dipindai

1. Buka [halaman generator IAC](#) CloudFormation konsol.
2. Pada bilah navigasi di bagian atas layar, pilih Wilayah AWS yang berisi sumber daya yang dipindai.
3. Dari bagian Template, pilih Buat template.
4. Pilih Mulai dari template baru.
 - a. Untuk nama Template, berikan nama untuk template Anda.

- b. (Opsional) Konfigurasi kebijakan Penghapusan Anda dan Perbarui kebijakan penggantian.
 - c. Pilih Berikutnya untuk menambahkan sumber daya yang dipindai ke template.
5. Untuk Tambahkan sumber daya yang dipindai, telusuri daftar sumber daya yang dipindai dan pilih sumber daya yang ingin Anda tambahkan ke templat Anda. Anda dapat memfilter sumber daya berdasarkan pengenalan sumber daya, jenis sumber daya, atau tag. Filter saling inklusif.
6. Ketika Anda telah menambahkan semua sumber daya yang diperlukan ke template Anda, pilih Berikutnya untuk keluar dari halaman Tambahkan sumber daya yang dipindai dan lanjutkan ke halaman Tambahkan sumber daya terkait.
7. Tinjau daftar sumber daya terkait yang direkomendasikan. Sumber daya terkait, seperti EC2 instans Amazon dan grup keamanan, saling bergantung dan biasanya termasuk dalam beban kerja yang sama. Pilih sumber daya terkait yang ingin Anda sertakan dalam template yang dihasilkan.

 Note

Kami menyarankan Anda menambahkan semua sumber daya terkait ke template ini.

8. Tinjau detail template, sumber daya yang dipindai, dan sumber daya terkait.
9. Pilih Buat template untuk keluar dari halaman Review dan buat dan buat template.

Buat template dari sumber daya yang dipindai ()AWS CLI

Untuk membuat template tumpukan dari sumber daya yang dipindai

1. Gunakan [list-resource-scan-resources](#) perintah untuk membuat daftar sumber daya yang ditemukan selama pemindaian, secara opsional menentukan `--resource-identifier` opsi untuk membatasi output. Untuk `--resource-scan-id` opsi, ganti sampel ARN dengan ARN yang sebenarnya.

```
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
```

Berikut ini adalah contoh respon, di mana `ManagedByStack` menunjukkan apakah CloudFormation mengelola sumber daya sudah. Salin outputnya. Anda membutuhkannya untuk langkah selanjutnya.

```
{
  "Resources": [
    {
      "ResourceType": "AWS::EKS::Cluster",
      "ResourceIdentifier": {
        "ClusterName": "MyAppClusterName"
      },
      "ManagedByStack": false
    },
    {
      "ResourceType": "AWS::AutoScaling::AutoScalingGroup",
      "ResourceIdentifier": {
        "AutoScalingGroupName": "MyAppASGName"
      },
      "ManagedByStack": false
    }
  ]
}
```

Untuk deskripsi bidang dalam output, lihat [ScannedResource](#) di Referensi AWS CloudFormation API.

- Gunakan cat perintah untuk menyimpan jenis sumber daya dan pengidentifikasi dalam file JSON bernama `resources.json` di direktori home Anda. Berikut ini adalah contoh JSON berdasarkan contoh output pada langkah sebelumnya.

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

```

        "AutoScalingGroupName": "MyAppASGName"
    }
}
]

```

- Gunakan perintah [list-resource-scan-related-resources](#), bersama dengan `resources.json` file yang Anda buat, untuk membuat daftar sumber daya yang terkait dengan sumber daya yang dipindai.

```

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

```

Berikut ini adalah contoh respon, di mana `ManagedByStack` menunjukkan apakah CloudFormation mengelola sumber daya sudah. Tambahkan sumber daya ini ke file JSON yang Anda buat di langkah sebelumnya. Anda akan membutuhkannya untuk membuat template Anda.

```

{
  "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
    }
  ]
}

```

Untuk deskripsi bidang dalam output, lihat [ScannedResource](#) di Referensi AWS CloudFormation API.

 Note

Daftar input sumber daya tidak dapat melebihi panjang 100. Untuk membuat daftar sumber daya terkait untuk lebih dari 100 sumber daya, jalankan `list-resource-scan-related-resources` perintah dalam batch 100 dan konsolidasikan hasilnya. Ketahuilah bahwa output mungkin berisi sumber daya duplikat dalam daftar.

4. Gunakan [create-generated-template](#) perintah untuk membuat template tumpukan baru, sebagai berikut, dengan modifikasi ini:

- Ganti `us-east-1` dengan Wilayah AWS yang berisi sumber daya yang dipindai.
- Ganti `MyTemplate` dengan nama template yang akan dibuat.

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

Berikut ini adalah contoh `resources.json` file.

```
[  
  {  
    "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"
      }
    }
  ]
}
```

Jika berhasil, perintah ini mengembalikan berikut ini.

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

Buat CloudFormation tumpukan dari sumber daya yang dipindai

Setelah Anda membuat template Anda, Anda dapat melihat pratinjau template yang dihasilkan dengan Infrastructure Composer sebelum membuat tumpukan dan mengimpor sumber daya yang dipindai. Ini membantu Anda memvisualisasikan arsitektur aplikasi lengkap dengan sumber daya dan hubungannya. Untuk informasi selengkapnya tentang Infrastructure Composer, lihat [Buat template secara visual dengan Infrastructure Composer](#).

Untuk membuat tumpukan dan mengimpor sumber daya yang dipindai

1. Buka [halaman generator IAC](#) CloudFormation konsol.
2. Pada bilah navigasi di bagian atas layar, pilih Wilayah AWS untuk template Anda.
3. Pilih tab Template, lalu pilih nama template Anda untuk melihat informasi selengkapnya.
4. Pada tab Definisi Template, di bagian atas bagian Template, Anda dapat mengganti template dari YAMAL ke sintaks JSON berdasarkan preferensi Anda.
5. Tinjau detail template Anda untuk memastikan semuanya sudah diatur dengan benar. Untuk membuatnya lebih mudah untuk meninjau dan memahami template, Anda dapat beralih dari tampilan kode default ke tampilan grafis dari infrastruktur yang dijelaskan dalam template

menggunakan Infrastructure Composer. Untuk melakukannya, di bawah Template, pilih Canvas bukan Template.

Tindakan kanvas

- Untuk fokus pada detail sumber daya tertentu dalam template Anda, klik dua kali kartu untuk membuka panel Resource properties.
 - Untuk mengatur dan mengatur kartu secara visual di kanvas, pilih Atur dari kiri atas kanvas.
 - Untuk memperbesar dan memperkecil kanvas Anda, gunakan kontrol zoom di kanan bawah kanvas.
6. Untuk melihat sumber daya tertentu di konsol, pilih tab Sumber daya Template, lalu pilih ID fisik sumber daya yang ingin Anda lihat. Ini membawa Anda ke konsol untuk sumber daya tertentu. Anda juga dapat menambahkan, menghapus, dan menyinkronkan ulang sumber daya dalam definisi templat dari tab Sumber daya Template.
 7. Pada tab Definisi Template, generator IAC mungkin mengeluarkan peringatan tentang sumber daya yang berisi properti khusus tulis. Setelah meninjau peringatan, Anda dapat mengunduh templat yang dihasilkan dan membuat perubahan yang diperlukan. Untuk informasi selengkapnya, lihat [Selesaikan properti hanya tulis](#).
 8. Ketika Anda puas dengan definisi template Anda, pada tab Definisi template, pilih Impor untuk ditumpuk dan kemudian pilih Berikutnya.
 9. Pada panel Tentukan tumpukan dari halaman Tentukan detail tumpukan, masukkan nama tumpukan Anda, lalu pilih Berikutnya.
 10. Tinjau dan masukkan parameter untuk tumpukan. Pilih Berikutnya.
 11. Tinjau opsi Anda di halaman Tinjau perubahan dan pilih Berikutnya.
 12. Tinjau detail Anda di halaman Tinjau dan impor dan pilih Impor sumber daya.

Selesaikan properti hanya tulis

Dengan generator CloudFormation IAC, Anda dapat membuat template menggunakan sumber daya yang disediakan di akun Anda yang belum dikelola oleh CloudFormation. Namun, properti sumber daya tertentu ditetapkan sebagai write-only, yang berarti mereka dapat ditulis tetapi tidak dapat dibaca oleh CloudFormation, misalnya, kata sandi database.

Saat membuat CloudFormation templat dari sumber daya yang ada, properti khusus tulis menimbulkan tantangan. Dalam kebanyakan kasus, CloudFormation mengubah properti ini menjadi parameter dalam template yang dihasilkan. Ini memungkinkan Anda untuk memasukkan properti

sebagai nilai parameter selama operasi impor. Namun, ada skenario di mana konversi ini tidak mungkin, dan CloudFormation menangani kasus ini secara berbeda.

Properti yang saling eksklusif

Beberapa sumber daya memiliki beberapa set properti yang saling eksklusif, setidaknya beberapa di antaranya hanya ditulis. Dalam kasus ini, generator IAc tidak dapat menentukan kumpulan properti eksklusif mana yang diterapkan ke sumber daya selama pembuatan. Misalnya, Anda dapat memberikan kode untuk [AWS::Lambda::Function](#) menggunakan salah satu set properti ini.

- Code/S3Bucket, Code/S3Key, dan secara opsional Code/S3ObjectVersion
- Code/ImageUri
- Code/ZipFile

Semua properti ini hanya ditulis. Generator IAc memilih salah satu set properti eksklusif dan menambahkannya ke template yang dihasilkan. Parameter ditambahkan untuk masing-masing properti write-only. Nama parameter termasuk OneOf dan deskripsi parameter menunjukkan bahwa properti yang sesuai dapat diganti dengan properti eksklusif lainnya. Generator IAc menetapkan jenis peringatan MUTUALLY_EXCLUSIVE_PROPERTIES untuk properti yang disertakan.

Jenis yang saling eksklusif

Dalam beberapa kasus, properti write-only dapat terdiri dari beberapa tipe data. Misalnya, Body properti [AWS::ApiGateway::RestApi](#) dapat berupa object atau astring. Ketika ini terjadi, generator IAc menyertakan properti dalam template yang dihasilkan menggunakan jenis string dan menetapkan jenis MUTUALLY_EXCLUSIVE_TYPES peringatan.

Arrayproperti

Jika properti write-only memiliki tipe `array`, generator IAc tidak dapat memasukkannya ke dalam template yang dihasilkan karena parameter hanya dapat berupa nilai skalar. Dalam hal ini, properti dihilangkan dari template, dan jenis peringatan UNSUPPORTED_PROPERTIES ditetapkan.

Properti opsional

Untuk properti write-only opsional, generator IAc tidak dapat mendeteksi apakah properti digunakan saat menyiapkan sumber daya. Dalam hal ini, properti dihilangkan dari template yang dihasilkan, dan jenis peringatan UNSUPPORTED_PROPERTIES disetel.

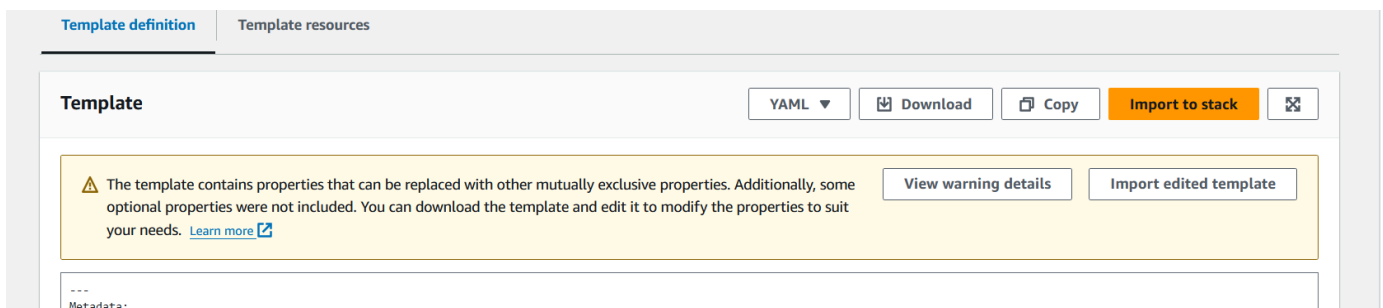
Peringatan dan langkah selanjutnya

Untuk menentukan properti mana yang hanya ditulis, Anda harus melihat peringatan yang dikembalikan oleh konsol generator IAc. [Referensi tipe AWS sumber daya dan properti](#) tidak menunjukkan apakah properti hanya ditulis, atau jika mendukung beberapa tipe.

Atau, Anda dapat melihat properti mana yang hanya ditulis dari skema penyedia sumber daya. Untuk mengunduh skema penyedia sumber daya, lihat [skema penyedia CloudFormation sumber daya](#).

Untuk mengatasi masalah dengan properti khusus tulis

1. Buka [halaman generator iAc](#) CloudFormation konsol.
2. Pada bilah navigasi di bagian atas layar, pilih Wilayah AWS untuk template Anda.
3. Pilih tab Template, lalu pilih nama template yang Anda buat.
4. Pada tab Definisi templat, ketika template yang dihasilkan menyertakan sumber daya dengan properti khusus tulis, konsol generator IAc menampilkan peringatan dengan ringkasan jenis masalah. Misalnya:



5. Pilih Lihat detail peringatan untuk detail selengkapnya. Sumber daya dengan properti write-only diidentifikasi oleh ID logis yang digunakan dalam template yang dihasilkan dan jenis sumber daya.

Gunakan daftar peringatan untuk mengidentifikasi sumber daya dengan properti hanya tulis dan lihat setiap sumber daya untuk menentukan perubahan apa (jika ada) yang perlu dilakukan pada templat yang dihasilkan.

Resource warnings (1)

▼ LambdaFunction001aClambda002fd6K (AWS::Lambda::Function) [Learn more](#)

Unsupported properties

1. SnapStart/ApplyOn

Applying SnapStart setting on function resource type. Possible values: [PublishedVersions, None] This property can be replaced with other types

2. Code/S3ObjectVersion

For versioned objects, the version of the deployment package object to use. This property can be replaced with other exclusive properties

Mutually exclusive properties

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/S3Key

The Amazon S3 key of the deployment package. This property can be replaced with other exclusive properties

6. Jika template Anda harus diperbarui untuk menyelesaikan masalah dengan properti write-only, lakukan hal berikut:
 - a. Pilih Unduh untuk mengunduh salinan templat.
 - b. Edit template Anda.
 - c. Saat perubahan selesai, Anda dapat memilih tombol Impor templat yang diedit untuk melanjutkan proses impor.

Cara mengatasi masalah dengan properti khusus tulis di sumber daya AWS::ApiGateway::RestAPI

Topik ini menjelaskan cara mengatasi masalah dengan properti khusus tulis di [AWS::ApiGateway::RestApi](#) sumber daya saat menggunakan generator IAc.

Isu

Ketika template yang dihasilkan berisi `AWS::ApiGateway::RestApi` sumber daya, maka peringatan dihasilkan yang menyatakan bahwa `Body`, `BodyS3Location`, dan `CloneFrom` properti diidentifikasi sebagai `UNSUPPORTED_PROPERTIES`. Ini karena ini adalah properti write-only opsional. Generator IAc tidak tahu apakah properti ini pernah diterapkan pada sumber daya. Oleh karena itu, ia menghilangkan properti ini dalam template yang dihasilkan.

Resolusi

Untuk menyetel `Body` properti untuk REST API Anda, perbarui template yang Anda buat.

1. Gunakan tindakan API Amazon [GetExport](#) API Gateway untuk mengunduh API. Misalnya, dengan menggunakan [aws apigateway get-export](#) AWS CLI perintah. Untuk informasi

selengkapnya, lihat [Mengeksport REST API dari API Gateway](#) di Panduan Pengembang API Gateway.

2. Ambil Body properti dari respons tindakan GetExport API. Unggah ke bucket Amazon S3.
3. Unduh template yang dihasilkan.
4. Tambahkan BodyS3Location/Key properti BodyS3Location/Bucket dan ke template, tentukan nama bucket dan kunci tempat penyimpanan. Body
5. Buka template yang dihasilkan di konsol generator IAC dan pilih Impor template yang diedit.

Cara mengatasi masalah dengan properti khusus tulis di sumber daya AWS::Lambda::Function

Topik ini menjelaskan cara mengatasi masalah dengan properti khusus tulis di [AWS::Lambda::Function](#) sumber daya saat menggunakan generator IAC.

Isu

Sumber `AWS::Lambda::Function` daya memiliki tiga set properti yang saling eksklusif untuk menentukan kode Lambda:

- `Code/S3Bucket` dan `Code/S3Key` properti, dan opsional properti `Code/S3ObjectVersion`
- `Code/ImageUri` properti
- `Code/ZipFile` properti

Hanya satu dari set ini yang dapat digunakan untuk `AWS::Lambda::Function` sumber daya tertentu.

Generator IAC tidak dapat menentukan kumpulan properti khusus tulis eksklusif mana yang digunakan untuk membuat atau memperbarui sumber daya. Akibatnya, ini hanya mencakup set properti pertama dalam template yang dihasilkan. `Code/ZipFile` properti `Code/ImageUri` dan dihilangkan.

Selain itu, generator IAC mengeluarkan peringatan berikut:

- **MUTUALLY_EXCLUSIVE_PROPERTIES** Memperingatkan bahwa `Code/S3Bucket` dan `Code/S3Key` diidentifikasi sebagai properti yang saling eksklusif.
- **UNSUPPORTED_PROPERTIES** Memperingatkan bahwa `Code/S3ObjectVersion` properti tidak didukung.

Untuk menyertakan `AWS::Lambda::Function` sumber daya dalam template yang dihasilkan, Anda harus mengunduh dan memperbarui template dengan properti kode yang benar.

Resolusi

Jika Anda menyimpan kode Lambda di bucket Amazon S3 dan tidak menggunakan **S3ObjectVersion** properti, Anda dapat mengimpor template yang dihasilkan tanpa modifikasi apa pun. Generator IAC akan meminta Anda untuk bucket Amazon S3 dan kunci sebagai parameter template selama operasi impor.

Jika Anda menyimpan kode Lambda Anda sebagai repositori Amazon ECR, Anda dapat memperbarui template Anda menggunakan petunjuk berikut:

1. Unduh template yang dihasilkan.
2. Hapus properti dan parameter yang sesuai untuk `Code/S3Key` properti `Code/S3Bucket` dan dari template yang dihasilkan.
3. Ganti properti yang dihapus dalam template yang dihasilkan dengan `Code/ImageUri` properti, tentukan URL untuk repositori Amazon ECR.
4. Buka template yang dihasilkan di konsol generator IAC dan pilih tombol Impor templat yang diedit.

Jika Anda menyimpan kode Lambda Anda seperti dalam file zip, Anda dapat memperbarui template Anda menggunakan petunjuk berikut:

1. Unduh template yang dihasilkan.
2. Hapus properti dan parameter yang sesuai untuk `Code/S3Key` properti `Code/S3Bucket` dan dari template yang dihasilkan.
3. Ganti properti yang dihapus dalam template yang dihasilkan dengan `Code/ZipFile` properti.
4. Buka template yang dihasilkan di konsol generator IAC dan pilih tombol Impor templat yang diedit.

Jika Anda tidak memiliki salinan kode Lambda Anda, Anda dapat memperbarui template Anda menggunakan petunjuk berikut:

1. Gunakan tindakan AWS Lambda [GetFunction](#) API (misalnya, dengan menggunakan [aws lambda get-function](#) AWS CLI perintah).

2. Sebagai tanggapan, `RepositoryType` parameternya adalah `S3` jika kode ada di bucket Amazon S3, atau `ECR` jika kode tersebut ada di repositori Amazon ECR.
3. Sebagai tanggapan, `Location` parameter berisi URL yang telah ditandatangani sebelumnya yang dapat Anda gunakan untuk mengunduh paket penerapan selama 10 menit. Unduh kodenya.
4. Unggah kode ke bucket Amazon S3.
5. Jalankan operasi impor dengan templat yang dihasilkan dan berikan nama dan kunci bucket sebagai nilai parameter.

Dapatkan nilai yang disimpan di layanan lain menggunakan referensi dinamis

Referensi dinamis memberikan cara yang nyaman bagi Anda untuk menentukan nilai eksternal yang disimpan dan dikelola di layanan lain dan memisahkan informasi sensitif dari templat Anda `infrastructure-as-code`. CloudFormation mengambil nilai referensi yang ditentukan bila diperlukan selama tumpukan dan mengubah operasi set.

Dengan referensi dinamis, Anda dapat:

- Gunakan string aman — Untuk data sensitif, selalu gunakan parameter string aman di AWS Systems Manager Parameter Store atau rahasia AWS Secrets Manager untuk memastikan data Anda dienkripsi saat istirahat.
- Batasi akses — Batasi akses ke parameter Parameter Store atau rahasia Secrets Manager hanya untuk kepala sekolah dan peran yang berwenang.
- Putar kredensial — Putar data sensitif Anda secara teratur yang disimpan di Parameter Store atau Secrets Manager untuk mempertahankan tingkat keamanan yang tinggi.
- Rotasi otomatis — Manfaatkan fitur rotasi otomatis Secrets Manager untuk memperbarui dan mendistribusikan data sensitif Anda secara berkala di seluruh aplikasi dan lingkungan Anda.

Pertimbangan umum

Berikut ini adalah pertimbangan umum untuk Anda pertimbangkan sebelum Anda menentukan referensi dinamis dalam CloudFormation templat Anda:

- Hindari menyertakan referensi dinamis, atau data sensitif apa pun, di properti sumber daya yang merupakan bagian dari pengenalan utama sumber daya. CloudFormation dapat menggunakan nilai plaintext aktual dalam pengidentifikasi sumber daya utama, yang bisa menjadi risiko keamanan. ID sumber daya ini mungkin muncul dalam output turunan atau tujuan.

Untuk menentukan properti sumber daya mana yang terdiri dari pengidentifikasi utama tipe sumber daya, lihat dokumentasi referensi sumber daya untuk sumber daya tersebut dalam referensi [tipe AWS sumber daya dan properti](#). Di bagian Kembalikan nilai, nilai Ref pengembalian fungsi mewakili properti sumber daya yang terdiri dari tipe pengidentifikasi utama sumber daya.

- Anda dapat menyertakan hingga 60 referensi dinamis dalam templat tumpukan.
- Jika Anda menggunakan transformasi (seperti `AWS::Include` atau `AWS::Serverless`), CloudFormation tidak menyelesaikan referensi dinamis sebelum menerapkan transformasi. Sebagai gantinya, ia meneruskan string literal dari referensi dinamis ke transformasi, dan menyelesaikan referensi saat Anda menjalankan set perubahan menggunakan template.
- Referensi dinamis tidak dapat digunakan untuk nilai aman (seperti yang disimpan di Parameter Store atau Secrets Manager) di sumber daya khusus.
- Referensi dinamis juga tidak didukung dalam `AWS::CloudFormation::Init` metadata dan properti Amazon EC2 UserData.
- Jangan membuat referensi dinamis yang diakhiri dengan garis miring terbalik (`\`). CloudFormation tidak dapat menyelesaikan referensi ini, yang akan menyebabkan operasi tumpukan gagal.

Topik berikut memberikan informasi dan pertimbangan lain untuk menggunakan referensi dinamis.

Topik

- [Dapatkan nilai plaintext dari Systems Manager Parameter Store](#)
- [Dapatkan nilai string aman dari Systems Manager Parameter Store](#)
- [Dapatkan nilai rahasia atau rahasia dari Secrets Manager](#)

Dapatkan nilai plaintext dari Systems Manager Parameter Store

Saat membuat CloudFormation template, Anda mungkin ingin menggunakan nilai plaintext yang disimpan di Parameter Store. Parameter Store adalah kemampuan AWS Systems Manager. Untuk pengantar Parameter Store, lihat [AWS Systems Manager Parameter Menyimpan](#) di Panduan AWS Systems Manager Pengguna.

Untuk menggunakan nilai plaintext dari Parameter Store dalam template Anda, Anda menggunakan referensi `ssm` dinamis. Referensi ini memungkinkan Anda untuk mengakses nilai dari parameter tipe `String` atau `StringList` di Parameter Store.

Untuk memverifikasi versi referensi `ssm` dinamis mana yang akan digunakan dalam operasi tumpukan, buat set perubahan untuk operasi tumpukan. Kemudian, tinjau template yang diproses pada tab Template. Untuk informasi selengkapnya, lihat [Buat set perubahan untuk CloudFormation tumpukan](#).

Saat menggunakan referensi `ssm` dinamis, ada beberapa hal penting yang perlu diingat:

- CloudFormation tidak mendukung deteksi drift pada referensi dinamis. Untuk referensi `ssm` dinamis di mana Anda belum menentukan versi parameter, sebaiknya, jika Anda memperbarui versi parameter di Systems Manager, Anda juga melakukan operasi pembaruan tumpukan pada tumpukan apa pun yang menyertakan referensi `ssm` dinamis, untuk mengambil versi parameter terbaru.
- Untuk menggunakan referensi `ssm` dinamis di Parameters bagian CloudFormation template Anda, Anda harus menyertakan nomor versi. CloudFormation tidak memungkinkan Anda untuk mereferensikan nilai Parameter Store tanpa nomor versi di bagian ini. Atau, Anda dapat menentukan parameter Anda sebagai tipe parameter Systems Manager di template Anda. Ketika Anda melakukan ini, Anda dapat menentukan kunci parameter Systems Manager sebagai nilai default untuk parameter Anda. CloudFormation kemudian akan mengambil versi terbaru dari nilai parameter dari Parameter Store, tanpa Anda harus menentukan nomor versi. Ini dapat membuat template Anda lebih sederhana dan lebih mudah dirawat. Untuk informasi selengkapnya, lihat [Tentukan sumber daya yang ada saat runtime dengan CloudFormation tipe parameter yang disediakan](#).
- Untuk sumber daya kustom, CloudFormation selesaikan referensi `ssm` dinamis sebelum mengirim permintaan ke sumber daya kustom.
- CloudFormation tidak mendukung penggunaan referensi dinamis untuk mereferensikan parameter yang dibagikan dari yang lain Akun AWS.
- CloudFormation tidak mendukung penggunaan label parameter Systems Manager dalam referensi dinamis.

Izin

Untuk menentukan parameter yang disimpan di Systems Manager Parameter Store, Anda harus memiliki izin [GetParameters](#) untuk memanggil parameter yang ditentukan. Untuk mempelajari

cara membuat kebijakan IAM yang menyediakan akses ke parameter Systems Manager tertentu, lihat [Membatasi akses ke parameter Systems Manager menggunakan kebijakan IAM di Panduan Pengguna.AWS Systems Manager](#)

Pola referensi

Untuk mereferensikan nilai plaintext yang disimpan di Systems Manager Parameter Store di CloudFormation template Anda, gunakan pola ssm referensi berikut.

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

Referensi Anda harus mematuhi pola ekspresi reguler berikut untuk nama parameter dan versi:

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

parameter-name

Nama parameter dalam Penyimpanan Parameter. Perhatikan bahwa nama parameter bersifat peka huruf besar-kecil.

Diperlukan.

version

Integer yang menentukan versi parameter untuk digunakan. Jika Anda tidak menentukan versi yang tepat, CloudFormation gunakan versi terbaru dari parameter setiap kali Anda membuat atau memperbarui tumpukan. Untuk informasi selengkapnya, lihat [Bekerja dengan versi parameter](#) di Panduan AWS Systems Manager Pengguna.

Opsional.

Contoh

Contoh berikut membuat template EC2 peluncuran yang mereferensikan ID AMI kustom yang disimpan di Parameter Store. Referensi dinamis mengambil ID AMI *2* dari versi *golden-ami* parameter setiap kali instance diluncurkan dari template peluncuran.

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

```

Dapatkan nilai string aman dari Systems Manager Parameter Store

Di CloudFormation, Anda dapat menggunakan data sensitif seperti kata sandi atau kunci lisensi tanpa mengeksposnya langsung di templat Anda dengan menyimpan data sensitif sebagai “string aman” di AWS Systems Manager Parameter Store. Untuk pengenalan Parameter Store, lihat [AWS Systems Manager Parameter Menyimpan](#) di Panduan AWS Systems Manager Pengguna.

Untuk menggunakan string aman Parameter Store dalam template Anda, Anda menggunakan referensi `ssm-secure` dinamis. CloudFormation tidak pernah menyimpan nilai string aman yang sebenarnya. Sebaliknya, ia hanya menyimpan referensi dinamis literal, yang berisi nama parameter plaintext dari string aman.

Selama pembuatan tumpukan atau pembaruan, CloudFormation mengakses nilai string aman sesuai kebutuhan, tanpa mengekspos nilai sebenarnya. String aman hanya dapat digunakan untuk properti

sumber daya yang mendukung pola referensi `ssm-secure` dinamis. Untuk informasi selengkapnya, lihat [Sumber daya yang men-support pola parameter dinamis untuk string aman](#).

CloudFormation tidak mengembalikan nilai parameter aktual untuk string aman dalam panggilan API apa pun. Ini hanya mengembalikan referensi dinamis literal. Saat membandingkan perubahan menggunakan set perubahan, CloudFormation hanya membandingkan string referensi dinamis literal. Itu tidak menyelesaikan dan membandingkan nilai string aman yang sebenarnya.

Saat menggunakan referensi `ssm-secure` dinamis, ada beberapa hal penting yang perlu diingat:

- CloudFormation tidak dapat mengakses nilai Parameter Store dari yang lain Akun AWS.
- CloudFormation tidak mendukung penggunaan label parameter Systems Manager atau parameter publik dalam referensi dinamis.
- Di wilayah `cn-north-1` dan `cn-northwest-1` wilayah, string aman tidak didukung oleh Systems Manager.
- Referensi dinamis untuk nilai aman, seperti `ssm-secure`, saat ini tidak didukung dalam sumber daya khusus.
- Jika CloudFormation perlu memutar kembali pembaruan tumpukan, dan versi parameter string aman yang ditentukan sebelumnya tidak lagi tersedia, operasi rollback akan gagal. Dalam kasus seperti itu, Anda memiliki dua opsi:
 - Gunakan `CONTINUE_UPDATE_ROLLBACK` untuk melewati sumber daya.
 - Buat ulang parameter string aman di Systems Manager Parameter Store, dan perbarui hingga versi parameter mencapai versi yang digunakan dalam template. Kemudian, gunakan `CONTINUE_UPDATE_ROLLBACK` tanpa melewatkan sumber daya.

Sumber daya yang men-support pola parameter dinamis untuk string aman

Sumber daya yang mendukung pola referensi `ssm-secure` dinamis meliputi:

Sumber Daya	Jenis properti	Properti
AWS::DirectoryService::MicrosoftKLAN		Password
AWS::DirectoryService::SimpleKLAN		Password

Sumber Daya	Jenis properti	Properti
AWS::ElastiCache::ReplicationGroup		AuthToken
AWS::IAM::User	LoginProfile	Password
AWS::KinesisFirehose::DeliveryStream	RedshiftDestinationConfiguration	Password
AWS::OpsWorks::App	Sumber	Password
AWS::OpsWorks::Stack	CustomCookbooksSource	Password
AWS::OpsWorks::Stack	RdsDbInstances	DbPassword
AWS::RDS::DBCluster		MasterUserPassword
AWS::RDS::DBInstance		MasterUserPassword
AWS::Redshift::Cluster		MasterUserPassword

Pola referensi

Untuk mereferensikan nilai string aman dari Systems Manager Parameter Store di CloudFormation template Anda, gunakan pola ssm-secure referensi berikut.

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

Referensi Anda harus mematuhi pola ekspresi reguler berikut untuk nama parameter dan versi:

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

parameter-name

Nama parameter dalam Penyimpanan Parameter. Perhatikan bahwa nama parameter bersifat peka huruf besar-kecil.

Diperlukan.

version

Integer yang menentukan versi parameter untuk digunakan. Jika Anda tidak menentukan versi yang tepat, CloudFormation gunakan versi terbaru dari parameter setiap kali Anda membuat atau memperbarui tumpukan. Untuk informasi selengkapnya, lihat [Bekerja dengan versi parameter](#) di Panduan AWS Systems Manager Pengguna.

Opsional.

Contoh

Contoh berikut menggunakan referensi ssm-secure dinamis untuk mengatur kata sandi untuk pengguna IAM ke string aman yang disimpan di Parameter Store. Seperti yang ditentukan, CloudFormation akan menggunakan **10** versi ***IAMUserPassword*** parameter untuk tumpukan dan mengubah operasi set.

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}}'
```

Dapatkan nilai rahasia atau rahasia dari Secrets Manager

Secrets Manager adalah layanan yang memungkinkan Anda menyimpan dan mengelola rahasia dengan aman seperti kredensi database, kata sandi, dan kunci API pihak ketiga. Dengan

menggunakan Secrets Manager, Anda dapat menyimpan dan mengontrol akses ke rahasia ini secara terpusat, sehingga Anda dapat mengganti kredensi hardcode dalam kode Anda (termasuk kata sandi), dengan panggilan API ke Secrets Manager untuk mengambil rahasia secara terprogram. Untuk informasi lebih lanjut, lihat [Apa itu AWS Secrets Manager?](#) dalam AWS Secrets Manager User Guide.

Untuk menggunakan seluruh rahasia atau nilai rahasia yang disimpan di Secrets Manager dalam CloudFormation template Anda, Anda menggunakan referensi `secretsmanager` dinamis.

Pertimbangan

Saat menggunakan referensi `secretsmanager` dinamis, ada beberapa pertimbangan keamanan penting yang perlu diingat:

- Referensi `secretsmanager` dinamis dapat digunakan di semua atribut sumber daya. Menggunakan referensi `secretsmanager` dinamis menunjukkan bahwa Secrets Manager maupun CloudFormation log tidak boleh mempertahankan nilai rahasia yang diselesaikan. Namun, nilai rahasia dapat muncul di layanan yang sumber dayanya digunakan. Tinjau penggunaan Anda untuk menghindari kebocoran data rahasia.
- Memperbarui rahasia di Secrets Manager tidak secara otomatis memperbarui rahasia di CloudFormation. Untuk memperbarui referensi `secretsmanager` dinamis, Anda harus melakukan pembaruan tumpukan yang memperbarui sumber daya yang berisi referensi dinamis, baik dengan memperbarui properti sumber daya yang berisi referensi `secretsmanager` dinamis, atau memperbarui properti sumber daya lainnya.

Misalnya, Anda menentukan `MasterPassword` properti [AWS::RDS::DBInstance](#) sumber daya dalam templat untuk menjadi `secretsmanager` referensi dinamis, dan kemudian membuat tumpukan dari templat. Anda kemudian memperbarui nilai rahasia itu di Secrets Manager, tetapi jangan memperbarui `AWS::RDS::DBInstance` sumber daya di template Anda. Dalam kasus ini, bahkan jika Anda melakukan pembaruan tumpukan, nilai rahasia dalam `MasterPassword` properti tidak diperbarui, dan tetap nilai rahasia sebelumnya.

Juga, pertimbangkan untuk menggunakan Secrets Manager untuk memutar rahasia secara otomatis untuk layanan atau basis data aman. Untuk informasi selengkapnya, lihat [Merotasi rahasia AWS Secrets Manager](#).

- Referensi dinamis untuk nilai aman, seperti `secretsmanager`, saat ini tidak didukung dalam sumber daya khusus.

Izin

Untuk menentukan rahasia yang disimpan di Secrets Manager, Anda harus memiliki izin [GetSecretValue](#) untuk memanggil rahasia tersebut.

Pola referensi

Untuk mereferensikan rahasia Secrets Manager di CloudFormation template Anda, gunakan pola `secretsmanager` referensi berikut.

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

`secret-id`

Nama atau ARN rahasianya.

Untuk mengakses rahasia di Akun AWS, Anda hanya perlu menentukan nama rahasia. Untuk mengakses rahasia di tempat yang berbeda Akun AWS, tentukan ARN lengkap rahasia tersebut.

Wajib.

`secret-string`

Satu-satunya nilai yang di-support adalah `SecretString`. Nilai default-nya `SecretString`.

`json-key`

Nama kunci dari pasangan kunci-nilai yang nilainya ingin Anda ambil. Jika Anda tidak menentukan `json-key`, CloudFormation mengambil seluruh teks rahasia.

Segmen ini mungkin tidak memasukkan karakter titik dua (:).

`version-stage`

Label pementasan versi rahasia untuk digunakan. Secrets Manager menggunakan label pementasan untuk melacak versi yang berbeda selama proses rotasi. Jika Anda menggunakan `version-stage` maka jangan tentukan `version-id`. Jika Anda tidak menentukan salah satu `version-stage` atau `version-id`, maka defaultnya adalah `AWSCURRENT` versinya.

Segmen ini mungkin tidak memasukkan karakter titik dua (:).

version-id

Pengidentifikasi unik dari versi rahasia yang akan digunakan. Jika Anda menentukan `version-id`, jangan tentukan `version-stage`. Jika Anda tidak menentukan salah satu `version-stage` atau `version-id`, maka defaultnya adalah `AWSCURRENT` versinya.

Segmen ini mungkin tidak memasukkan karakter titik dua (:).

Contoh

Topik

- [Mengambil nilai nama pengguna dan kata sandi dari rahasia](#)
- [Mengambil seluruh SecretString](#)
- [Mengambil nilai dari versi rahasia tertentu](#)
- [Mengambil rahasia dari yang lain Akun AWS](#)

Mengambil nilai nama pengguna dan kata sandi dari rahasia

[AWS::RDS::DBInstance](#) Contoh berikut mengambil nama pengguna dan nilai kata sandi yang disimpan dalam *MySecret* rahasia. Versi rahasia yang diambil adalah versi dengan nilai tahap versi. `AWSCURRENT`

JSON

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


YAML

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

Mengambil seluruh SecretString

Referensi dinamis berikut mengambil SecretString for *MySecret*.

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

Atau:

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

Mengambil nilai dari versi rahasia tertentu

Referensi dinamis berikut mengambil *password* nilai untuk *AWSPREVIOUS* versi. *MySecret*

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

Mengambil rahasia dari yang lain Akun AWS

Referensi dinamis berikut mengambil SecretString untuk *MySecret* yang lain Akun AWS. Anda harus menentukan ARN rahasia lengkap untuk mengakses rahasia di tempat lain. Akun AWS

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

Referensi dinamis berikut mengambil *password* nilai untuk *MySecret* yang lain Akun AWS. Anda harus menentukan ARN rahasia lengkap untuk mengakses rahasia di tempat lain. Akun AWS

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

Dapatkan AWS nilai menggunakan parameter semu

Parameter semu adalah variabel bawaan yang menyediakan akses ke informasi AWS lingkungan penting seperti akun IDs, nama Wilayah, dan detail tumpukan yang dapat berubah antara penerapan atau lingkungan.

Anda dapat menggunakan parameter semu alih-alih nilai hard-code untuk membuat templat Anda lebih portabel dan lebih mudah digunakan kembali di berbagai wilayah dan wilayah. Akun AWS

Sintaksis

Anda dapat mereferensikan parameter semu menggunakan fungsi Ref intrinsik atau fungsi Fn::Sub.

Fn::Sub

Ref

Fungsi Ref intrinsik menggunakan sintaks umum berikut. Untuk informasi lebih lanjut, lihat [Ref](#).

JSON

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

YAML

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

Fn::Sub

Fungsi Fn::Sub intrinsik menggunakan format berbeda yang mencakup `${}` sintaks di sekitar parameter semu. Untuk informasi selengkapnya, lihat [Fn::Sub](#).

JSON

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

YAML

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

Parameter pseudo yang tersedia

AWS::AccountId

Mengembalikan Akun AWS ID akun di mana tumpukan sedang dibuat, seperti 123456789012.

Parameter semu ini biasanya digunakan saat mendefinisikan peran IAM, kebijakan, dan kebijakan sumber daya lainnya yang melibatkan spesifik akun. ARNs

AWS::NotificationARNs

Mengembalikan daftar Nama Sumber Daya Amazon (ARNs) untuk topik Amazon SNS yang menerima pemberitahuan peristiwa tumpukan. Anda dapat menentukan ini ARNs melalui --notification-arns opsi di AWS CLI atau melalui konsol saat Anda membuat atau memperbarui tumpukan Anda.

Tidak seperti parameter semu lainnya yang mengembalikan nilai tunggal, AWS::NotificationARNs mengembalikan daftar. ARNs Untuk mengakses ARN tertentu dalam daftar, gunakan fungsi Fn::Select intrinsik. Untuk informasi selengkapnya, lihat [Fn::Select](#).

AWS::NoValue

Menghapus atribut sumber daya yang sesuai jika ditentukan sebagai nilai kembali dalam fungsi intrinsik Fn::If. Untuk informasi selengkapnya, lihat [Fn::If](#).

Parameter semu ini sangat berguna untuk membuat properti sumber daya bersyarat yang seharusnya hanya disertakan dalam kondisi tertentu.

AWS::Partition

Mengembalikan partisi tempat sumber daya berada. Untuk standar Wilayah AWS, partisi adalah aws. Jika Anda memiliki sumber daya di partisi lain, maka partisi-nya adalah aws-*partitionname*. Misalnya, partisi untuk sumber daya di Wilayah Tiongkok (Beijing dan Ningxia) aws-cn adalah dan partisi untuk sumber daya di AWS GovCloud Wilayah (AS-Barat) adalah. aws-us-gov

Partisi merupakan bagian dari ARN untuk sumber daya. Menggunakan AWS::Partition memastikan template Anda bekerja dengan benar di berbagai AWS partisi.

AWS::Region

Mengembalikan string yang mewakili Wilayah tempat sumber daya yang dicakup sedang dibuat, seperti `us-west-2`.

Ini adalah salah satu parameter semu yang paling umum digunakan, karena memungkinkan template untuk beradaptasi dengan yang berbeda Wilayah AWS tanpa modifikasi.

AWS::StackId

Mengembalikan ID (ARN) dari tumpukan, seperti. `arn:aws:cloudformation:us-west-2:123456789012:stack/teststack/51af3dc0-da77-11e4-872e-1234567db123`

AWS::StackName

Mengembalikan nama tumpukan, seperti `teststack`.

Nama tumpukan biasanya digunakan untuk membuat nama sumber daya unik yang mudah diidentifikasi sebagai milik tumpukan tertentu.

AWS::URLSuffix

Mengembalikan akhiran untuk AWS domain di Wilayah AWS tempat tumpukan digunakan. Akhiran biasanya `amazonaws.com`, tetapi untuk Wilayah Tiongkok (Beijing), sufiksnya adalah `amazonaws.com.cn`

Parameter ini sangat berguna ketika membangun URLs untuk titik akhir AWS layanan.

Contoh

Topik

- [Penggunaan dasar](#)
- [Menggunakan AWS::NotificationARNs](#)
- [Properti bersyarat dengan AWS::NoValue](#)

Penggunaan dasar

Contoh berikut membuat dua sumber daya: topik Amazon SNS dan CloudWatch alarm yang mengirimkan pemberitahuan ke topik itu. Mereka menggunakan `AWS::StackName`, `AWS::Region`,

dan `AWS::AccountId` untuk secara dinamis memasukkan nama tumpukan, saat ini Wilayah AWS, dan ID akun ke dalam nama sumber daya, deskripsi, dan ARNs

JSON

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

```

Menggunakan AWS::NotificationARNs

Contoh berikut mengonfigurasi grup Auto Scaling untuk mengirim notifikasi misalnya peristiwa peluncuran dan kesalahan peluncuran. Konfigurasi menggunakan parameter `AWS::NotificationARNs` semu, yang menyediakan daftar ARNs topik Amazon SNS yang ditentukan selama pembuatan tumpukan. `Fn::SelectFungsi` memilih ARN pertama dari daftar itu.

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
```

Properti bersyarat dengan AWS::NoValue

Contoh berikut membuat instans Amazon RDS DB yang menggunakan snapshot hanya jika ID snapshot disediakan. Jika UseDBSnapshot kondisi mengevaluasi ke true, CloudFormation menggunakan nilai DBSnapshotName parameter untuk DBSnapshotIdentifier properti. Jika kondisi dievaluasi menjadi false, CloudFormation hapus DBSnapshotIdentifier properti.

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

```

Dapatkan output yang diekspor dari tumpukan yang diterapkan CloudFormation

Jika Anda memiliki beberapa tumpukan di area yang sama Akun AWS dan Region, Anda mungkin ingin berbagi informasi di antara mereka. Ini berguna ketika satu tumpukan perlu menggunakan sumber daya yang dibuat oleh tumpukan lain.

Misalnya, Anda mungkin memiliki satu tumpukan yang membuat sumber daya jaringan, seperti subnet dan grup keamanan, untuk server web Anda. Tumpukan lain yang membuat server web yang sebenarnya kemudian dapat menggunakan sumber daya jaringan yang dibuat oleh tumpukan

pertama. Anda tidak perlu membuat sumber daya kode keras IDs di template tumpukan atau meneruskannya IDs sebagai parameter input.

Untuk berbagi informasi antar tumpukan, Anda mengekspor nilai keluaran dari satu tumpukan dan mengimpornya ke tumpukan lain. Begini cara kerjanya:

1. Di template tumpukan pertama (misalnya, tumpukan jaringan), Anda menentukan nilai tertentu untuk ekspor dengan menggunakan `Export` bidang di `Outputs` bagian tersebut. Untuk informasi selengkapnya, lihat [CloudFormation Outputssintaks template](#).
2. Saat Anda membuat atau memperbarui tumpukan itu, CloudFormation ekspor nilai output, membuatnya tersedia untuk tumpukan lain di wilayah Akun AWS dan yang sama.
3. Di template tumpukan lain, Anda menggunakan `Fn::ImportValue` fungsi untuk mengimpor nilai yang diekspor dari tumpukan pertama.
4. Saat Anda membuat atau memperbarui tumpukan kedua (misalnya, tumpukan server web), CloudFormation secara otomatis mengambil nilai yang diekspor dari tumpukan pertama dan menggunakannya.

Untuk panduan dan contoh templat, lihat [Lihat output sumber daya di tumpukan lain CloudFormation](#).

Mengekspor nilai output tumpukan versus menggunakan tumpukan bersarang

Tumpukan bersarang adalah tumpukan yang Anda buat dalam tumpukan lain dengan menggunakan `AWS::CloudFormation::Stack` sumber daya. Dengan tumpukan nest, Anda men-deploy dan mengelola semua sumber daya dari tumpukan tunggal. Anda dapat menggunakan output dari satu tumpukan dalam grup tumpukan nest sebagai input ke tumpukan lain dalam grup tersebut. Ini berbeda dari mengekspor nilai.

Jika Anda ingin mengisolasi berbagi informasi untuk dalam grup tumpukan nest, kami merekomendasikan Anda untuk menggunakan tumpukan nest. Untuk berbagi informasi dengan tumpukan lain (tidak hanya dalam grup tumpukan nest), ekspor nilai. Misalnya, Anda dapat membuat satu tumpukan dengan subnet dan kemudian mengekspor ID nya. Tumpukan lain dapat menggunakan subnet itu dengan mengimpor ID-nya. Setiap tumpukan tidak perlu membuat subnetnya sendiri. Selama tumpukan mengimpor ID subnet, Anda tidak dapat mengubah atau menghapusnya.

Untuk informasi lebih lanjut tentang tumpukan bersarang, lihat. [Pisahkan template menjadi potongan-potongan yang dapat digunakan kembali menggunakan tumpukan bersarang](#)

Pertimbangan

Pembatasan berikut berlaku untuk referensi cross-stack:

- Untuk masing-masing Akun AWS, `Export` nama harus unik dalam suatu Wilayah.
- Anda tidak dapat membuat referensi cross-stack di seluruh Wilayah. Anda dapat menggunakan fungsi intrinsik `Fn::ImportValue` untuk mengimpor hanya nilai yang telah diekspor dalam Wilayah yang sama.
- Untuk output, nilai `Name` properti tidak `Export` dapat digunakan `Ref` atau `GetAtt` fungsi yang bergantung pada sumber daya.

Demikian pula, `ImportValue` fungsi tidak dapat menyertakan `Ref` atau `GetAtt` fungsi yang bergantung pada sumber daya.

- Setelah tumpukan lain mengimpor nilai output, Anda tidak dapat menghapus tumpukan yang mengekspor nilai output atau memodifikasi nilai output yang diekspor. Semua impor harus dihapus sebelum Anda dapat menghapus tumpukan ekspor atau mengubah nilai output.

Mencantumkan nilai output yang diekspor

Jika Anda perlu melihat nilai output yang diekspor dari tumpukan Anda, gunakan salah satu metode berikut:

Untuk mencantumkan nilai output yang diekspor (konsol)

1. Buka AWS CloudFormation konsol di <https://console.aws.amazon.com/cloudformation>.
2. Pada bilah navigasi di bagian atas layar, pilih Anda Wilayah AWS.
3. Dari panel navigasi kiri, pilih Ekspor.

Untuk mencantumkan nilai output yang diekspor (AWS CLI)

Gunakan perintah berikut [list-exports](#). Ganti `us-east-1` dengan Anda Wilayah AWS.

```
aws cloudformation list-exports --region us-east-1
```

Berikut ini adalah output contoh.

```
{
  "Exports": [
```

```
{
  "ExportingStackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/
private-vpc/99764070-b56c-xmpl-bee8-062a88d1d800",
  "Name": "private-vpc-subnet-a",
  "Value": "subnet-07b410xmplddcfa03"
},
{
  "ExportingStackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/
private-vpc/99764070-b56c-xmpl-bee8-062a88d1d800",
  "Name": "private-vpc-subnet-b",
  "Value": "subnet-075ed3xmplebd2fb1"
},
{
  "ExportingStackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/
private-vpc/99764070-b56c-xmpl-bee8-062a88d1d800",
  "Name": "private-vpc-vpcid",
  "Value": "vpc-011d7xmpl1100e9841"
}
]
```

CloudFormation menunjukkan nama dan nilai output yang diekspor untuk wilayah saat ini dan tumpukan tempat mereka diekspor. Untuk menggunakan nilai output yang diekspor di template tumpukan lain, Anda dapat mereferensikannya menggunakan nama ekspor dan `Fn::ImportValue` fungsinya.

Pencantuman tumpukan yang mengimpor nilai output yang diekspor

Untuk menghapus atau mengubah nilai output yang diekspor, Anda harus terlebih dahulu mengetahui tumpukan mana yang mengimpornya.

Untuk melihat tumpukan yang mengimpor nilai output yang diekspor, gunakan salah satu metode berikut:

Untuk mencantumkan tumpukan yang mengimpor nilai output yang diekspor (konsol)

1. Buka AWS CloudFormation konsol di <https://console.aws.amazon.com/cloudformation>.
2. Dari panel navigasi kiri, pilih Ekspor.
3. Untuk melihat tumpukan mana yang mengimpor nilai ekspor tertentu, pilih Nama Ekspor untuk nilai ekspor tersebut. CloudFormation menampilkan halaman detail ekspor, yang mencantumkan semua tumpukan yang mengimpor nilai.

Untuk mencantumkan tumpukan yang mengimpor nilai output yang diekspor (AWS CLI)

Gunakan perintah [list-imports](#). Ganti *us-east-1* dengan Anda Wilayah AWS dan *private-vpc-vpcid* dengan nama nilai output yang diekspor.

```
aws cloudformation list-imports --region us-east-1 \  
  --export-name private-vpc-vpcid
```

CloudFormation mengembalikan daftar tumpukan yang mengimpor nilai.

```
{  
  "Imports": [  
    "my-app-stack"  
  ]  
}
```

Setelah Anda mengetahui tumpukan mana yang mengimpor nilai ekspor tertentu, Anda perlu memodifikasi tumpukan tersebut untuk menghapus `Fn::ImportValue` fungsi yang mereferensikan nilai output. Anda harus menghapus semua impor yang mereferensikan nilai output yang diekspor sebelum Anda dapat menghapus atau mengubah nilai output yang diekspor.

Tentukan sumber daya yang ada saat runtime dengan CloudFormation tipe parameter yang disediakan

Saat membuat templat, Anda dapat membuat parameter yang mengharuskan pengguna memasukkan pengenalan AWS sumber daya yang ada atau parameter Systems Manager dengan menggunakan tipe parameter khusus yang disediakan oleh CloudFormation.

Topik

- [Gambaran Umum](#)
- [Contoh](#)
- [Pertimbangan](#)
- [Jenis parameter AWS spesifik yang didukung](#)
- [Jenis parameter Systems Manager yang Didukung](#)
- [Tipe parameter Systems Manager yang tidak didukung](#)

Gambaran Umum

Di CloudFormation, Anda dapat menggunakan parameter untuk menyesuaikan tumpukan Anda dengan memberikan nilai input selama pembuatan atau pembaruan tumpukan. Fitur ini membuat template Anda dapat digunakan kembali dan fleksibel di berbagai skenario.

Parameter didefinisikan di `Parameters` bagian CloudFormation template. Setiap parameter memiliki nama dan tipe, dan dapat memiliki pengaturan tambahan seperti nilai default dan nilai yang diizinkan. Untuk informasi selengkapnya, lihat [CloudFormation Parameters sintaks template](#).

Jenis parameter menentukan jenis nilai input yang dapat diterima parameter. Misalnya, `Number` hanya menerima nilai numerik, sementara `String` menerima input teks.

CloudFormation menyediakan beberapa jenis parameter tambahan yang dapat Anda gunakan dalam template Anda untuk mereferensikan AWS sumber daya yang ada dan parameter Systems Manager.

Jenis parameter ini terbagi dalam dua kategori:

- **AWS Jenis parameter spesifik** - CloudFormation menyediakan satu set tipe parameter yang membantu catch nilai yang tidak valid saat membuat atau memperbarui tumpukan. Saat Anda menggunakan tipe parameter ini, siapa pun yang menggunakan template Anda harus menentukan nilai yang valid dari Akun AWS dan Wilayah tempat mereka membuat tumpukan.

Jika mereka menggunakan AWS Management Console, CloudFormation berikan daftar nilai yang ada yang sudah terisi sebelumnya dari akun dan Wilayah mereka. Dengan cara ini, pengguna tidak perlu mengingat dan mengetikkan nama atau ID tertentu dengan benar. Sebaliknya, mereka hanya memilih nilai dari daftar drop-down. Dalam beberapa kasus, mereka bahkan dapat mencari nilai berdasarkan ID, nama, atau nilai Name tag.

- **Jenis parameter Systems Manager** — CloudFormation juga menyediakan tipe parameter yang sesuai dengan parameter yang ada di Systems Manager Parameter Store. Bila Anda menggunakan tipe parameter ini, siapa pun yang menggunakan template Anda harus menentukan kunci Parameter Store sebagai nilai dari tipe parameter Systems Manager, dan CloudFormation kemudian mengambil nilai terbaru dari Parameter Store untuk digunakan dalam tumpukan mereka. Ini dapat berguna ketika Anda perlu sering memperbarui aplikasi dengan nilai properti baru, seperti Amazon Machine Image (AMI) baru IDs. Untuk informasi tentang Parameter Store, lihat [Systems Manager Parameter Store](#).

Setelah parameter Anda ditentukan di `Parameters` bagian, Anda dapat mereferensikan nilai parameter di seluruh CloudFormation template Anda menggunakan `Ref` fungsi tersebut.

Contoh

Contoh berikut menunjukkan template yang menggunakan jenis parameter berikut.

- `AWS::EC2::VPC::Id`
- `AWS::EC2::Subnet::Id`
- `AWS::EC2::KeyPair::KeyName`
- `AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>`

Untuk membuat tumpukan dari template ini, Anda harus menentukan ID VPC, ID subnet, dan nama key pair yang ada dari akun Anda. Anda juga dapat menentukan kunci Parameter Store yang ada yang mereferensikan ID AMI yang diinginkan atau mempertahankan nilai default `/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2`. Parameter publik ini adalah alias untuk ID AMI regional untuk AMI Amazon Linux 2 terbaru. Untuk informasi selengkapnya tentang parameter publik, lihat [Menemukan parameter publik di Penyimpanan Parameter](#) di Panduan AWS Systems Manager Pengguna.

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 perintah untuk membuat tumpukan

Berikut ini [create-stack](#) perintah membuat tumpukan berdasarkan contoh template.


```
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"
```

Untuk menggunakan tipe parameter yang menerima daftar string, seperti `List<AWS::EC2::Subnet::Id>`, Anda harus keluar dari koma di dalam `ParameterValue` dengan garis miring terbalik ganda, seperti yang ditunjukkan pada contoh berikut.

```
--parameters ParameterKey="SubnetIDs",ParameterValue="subnet-5ea0c127\\, subnet-6194ea3b  
\\, subnet-c87f2be0"
```

Pertimbangan

Sangat disarankan agar Anda menggunakan referensi dinamis untuk membatasi akses ke definisi konfigurasi sensitif, seperti kredensi pihak ketiga. Untuk informasi selengkapnya, lihat [Dapatkan nilai yang disimpan di layanan lain menggunakan referensi dinamis](#).

Jika Anda ingin mengizinkan pengguna template untuk menentukan nilai dari yang berbeda Akun AWS, jangan gunakan tipe parameter AWS-specific. Sebaliknya, tentukan parameter tipe `String` atau `CommaDelimitedList`.

Ada beberapa hal yang perlu diingat dengan tipe parameter Systems Manager:

- Anda dapat melihat nilai parameter yang diselesaikan pada tab Parameter tumpukan di konsol, atau dengan menjalankan [describe-stacks](#) atau [describe-change-set](#). Ingat, nilai-nilai ini ditetapkan ketika tumpukan dibuat atau diperbarui, sehingga mereka mungkin berbeda dari nilai terbaru di Parameter Store.
- Untuk pembaruan tumpukan, saat Anda menggunakan opsi Use existing value (atau disetel `UsePreviousValue` ke `true`), ini berarti Anda ingin tetap menggunakan kunci Parameter Store yang sama, bukan nilainya. CloudFormation selalu mengambil nilai terbaru.
- Jika Anda menentukan nilai yang diizinkan atau kendala lainnya, CloudFormation validasi terhadap kunci parameter yang Anda tentukan, tetapi bukan nilainya. Anda harus memvalidasi nilai di Parameter Store itu sendiri.

- Saat Anda membuat atau memperbarui tumpukan dan membuat set perubahan, CloudFormation gunakan nilai apa pun yang ada di Parameter Store pada saat itu. Jika parameter tertentu tidak ada di Parameter Store di bawah pemanggil Akun AWS, CloudFormation mengembalikan kesalahan validasi.
- Ketika Anda menjalankan set perubahan, CloudFormation menggunakan nilai-nilai yang ditentukan dalam set perubahan. Anda harus meninjau nilai-nilai ini sebelum mengeksekusi perubahan yang ditetapkan karena mereka mungkin berubah dalam Penyimpanan Parameter antara waktu saat Anda membuat perubahan set dan menjalankannya.
- Untuk parameter Parameter Store yang disimpan dalam hal yang sama Akun AWS, Anda harus memberikan nama parameter. Untuk parameter Parameter Store yang dibagikan oleh yang lain Akun AWS, Anda harus memberikan parameter ARN lengkap.

Jenis parameter AWS spesifik yang didukung

CloudFormation mendukung jenis AWS-spesifik berikut:

`AWS::EC2::AvailabilityZone::Name`

Availability Zone, seperti `us-west-2a`.

`AWS::EC2::Image::Id`

ID EC2 gambar Amazon, seperti `ami-0ff8a91507f77f867`. Perhatikan bahwa CloudFormation konsol tidak menampilkan daftar drop-down nilai untuk jenis parameter ini.

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

ID EC2 instans Amazon, seperti `i-1e731a32`.

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

Nama EC2 key pair Amazon.

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

Nama grup keamanan VPC default, seperti `my-sg-abc`

`AWS::EC2::SecurityGroup::Id`

ID grup keamanan, seperti `sg-a123fd85`.

`AWS::EC2::Subnet::Id`

ID subnet, seperti `subnet-123a351e`.

`AWS::EC2::Volume::Id`

ID volume Amazon EBS, seperti `vol-3cdd3f56`.

`AWS::EC2::VPC::Id`

ID VPC, seperti `vpc-a123baa3`.

`AWS::Route53::HostedZone::Id`

ID zona yang dihosting Amazon Route 53, seperti `Z23YXV40VPL04A`.

`List<AWS::EC2::AvailabilityZone::Name>`

Susunan Availability Zone untuk suatu wilayah, seperti `us-west-2a`, `us-west-2b`.

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

Array EC2 gambar Amazon IDs, seperti `ami-0ff8a91507f77f867`, `ami-0a584ac55a7631c0c`. Perhatikan bahwa CloudFormation konsol tidak menampilkan daftar drop-down nilai untuk jenis parameter ini.

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

Array EC2 contoh Amazon IDs, seperti `i-1e731a32`, `i-1e731a34`.

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

Array nama grup keamanan VPC default, seperti `my-sg-abc`, `my-sg-def`

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

Sebuah array kelompok keamanan IDs, seperti `sg-a123fd85`, `sg-b456fd85`.

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

Sebuah array subnet IDs, seperti `subnet-123a351e`, `subnet-456b351e`.

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

Array volume Amazon EBS IDs, seperti `vol-3cdd3f56`, `vol-4cdd3f56`.

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

Array VPC IDs, seperti `vpc-a123baa3`, `vpc-b456baa3`

`List<AWS::Route53::HostedZone::Id>`

Array zona yang dihosting Amazon Route 53 IDs, seperti `Z23YXV40VPL04A`, `Z23YXV40VPL04B`.

Jenis parameter Systems Manager yang Didukung

CloudFormation mendukung jenis parameter Systems Manager berikut:

`AWS::SSM::Parameter::Name`

Nama kunci parameter Systems Manager. Gunakan jenis parameter ini hanya untuk memeriksa apakah ada parameter yang diperlukan. CloudFormation tidak akan mengambil nilai aktual yang terkait dengan parameter.

`AWS::SSM::Parameter::Value<String>`

Parameter Systems Manager yang nilainya adalah string. Hal ini sesuai dengan `String` tipe parameter di Penyimpanan Parameter.

`AWS::SSM::Parameter::Value<List<String>>` atau

`AWS::SSM::Parameter::Value<CommaDelimitedList>`

Parameter Systems Manager yang nilainya adalah daftar string. Hal ini sesuai dengan `StringList` tipe parameter di Penyimpanan Parameter.

`AWS::SSM::Parameter::Value<AWS-specific parameter type>`

Parameter Systems Manager yang nilainya adalah tipe parameter AWS-spesifik.

Sebagai contoh, berikut menentukan `AWS::EC2::KeyPair::KeyName` tipe:

- `AWS::SSM::Parameter::Value<AWS::EC2::KeyPair::KeyName>`

`AWS::SSM::Parameter::Value<List<AWS-specific parameter type>>`

Parameter Systems Manager yang nilainya adalah daftar tipe parameter AWS-spesifik.

Sebagai contoh, berikut menentukan daftar `AWS::EC2::KeyPair::KeyName` tipe:

- `AWS::SSM::Parameter::Value<List<AWS::EC2::KeyPair::KeyName>>`

Tipe parameter Systems Manager yang tidak didukung

CloudFormation tidak mendukung tipe parameter Systems Manager berikut:

- Daftar tipe parameter Systems Manager — misalnya:
`List<AWS::SSM::Parameter::Value<String>>`

Selain itu, CloudFormation tidak mendukung mendefinisikan parameter template sebagai tipe parameter `SecureString` Systems Manager. Namun, Anda dapat menentukan string aman sebagai nilai parameter untuk sumber daya tertentu. Untuk informasi selengkapnya, lihat [Dapatkan nilai yang disimpan di layanan lain menggunakan referensi dinamis](#).

CloudFormation penelusuran

Dokumentasi ini menyediakan kumpulan panduan yang dirancang untuk memberi Anda praktik langsung dengan penerapan tumpukan.

- [Lihat output sumber daya di tumpukan lain CloudFormation](#)— Panduan ini menunjukkan kepada Anda cara mereferensikan output dari satu CloudFormation tumpukan dalam tumpukan lain. Alih-alih menyertakan semua sumber daya dalam satu tumpukan, Anda dapat membuat AWS sumber daya terkait di tumpukan terpisah untuk membuat templat yang lebih modular dan dapat digunakan kembali.
- [Rekan dengan VPC di yang lain Akun AWS](#)— Panduan ini memandu Anda melalui proses pembuatan koneksi peering Virtual Private Cloud (VPC) antara dua yang berbeda. VPCs Akun AWS Peering VPC membantu Anda merutekan lalu lintas antara VPCs dan mengakses sumber daya seolah-olah mereka adalah bagian dari jaringan yang sama.
- [Buat aplikasi yang diskalakan dan seimbang beban](#)— Temukan cara menggunakan CloudFormation untuk membuat aplikasi yang dapat diskalakan dan seimbang beban. Panduan ini mencakup pembuatan grup Auto Scaling, penyeimbang beban, dan sumber daya terkait lainnya untuk memastikan aplikasi Anda dapat menangani berbagai beban lalu lintas dan mempertahankan ketersediaan tinggi.
- [Menyebarkan aplikasi di Amazon EC2](#)— Pelajari cara menggunakan CloudFormation untuk menginstal, mengonfigurasi, dan memulai aplikasi secara otomatis di EC2 instans Amazon. Dengan cara ini, Anda dapat dengan mudah menduplikasi penerapan dan memperbarui instalasi yang ada tanpa menghubungkan langsung ke instance.
- [Memperbarui tumpukan](#)— Berjalanlah melalui perkembangan pembaruan sederhana ke tumpukan yang sedang berjalan dengan CloudFormation.
- [Lakukan blue/green penyebaran ECS melalui penggunaan CodeDeploy CloudFormation](#)— Temukan cara menggunakan CloudFormation untuk melakukan AWS CodeDeploy blue/green deployments on Amazon ECS. Blue/green penerapan adalah cara untuk memperbarui aplikasi atau layanan Anda dengan waktu henti minimal.

Lihat output sumber daya di tumpukan lain CloudFormation

Panduan ini menunjukkan kepada Anda cara mereferensikan output dari satu CloudFormation tumpukan dalam tumpukan lain untuk membuat templat yang lebih modular dan dapat digunakan kembali.

Alih-alih menyertakan semua sumber daya dalam satu tumpukan, Anda membuat AWS sumber daya terkait di tumpukan terpisah. Kemudian, Anda dapat merujuk ke output sumber daya yang diperlukan dari tumpukan lain. Dengan membatasi referensi lintas tumpukan untuk output, Anda mengontrol bagian dari tumpukan yang direferensikan oleh tumpukan lainnya.

Misalnya, Anda mungkin memiliki tumpukan jaringan dengan VPC, grup keamanan, dan subnet untuk aplikasi web publik, dan tumpukan aplikasi web publik terpisah. Untuk memastikan bahwa aplikasi web menggunakan grup keamanan dan subnet dari tumpukan jaringan, Anda membuat referensi lintas tumpukan yang memungkinkan tumpukan aplikasi web untuk mereferensi output sumber daya dari tumpukan jaringan. Dengan referensi lintas tumpukan, pemilik tumpukan aplikasi web tidak perlu membuat atau mempertahankan aturan atau aset jaringan.

Untuk membuat referensi lintas tumpukan, gunakan output bidang `Export` untuk menandai nilai output sumber daya untuk ekspor. Lalu, gunakan fungsi intrinsik `Fn::ImportValue` untuk mengimpor nilai. Untuk informasi selengkapnya, lihat [Dapatkan output yang diekspor dari tumpukan yang diterapkan CloudFormation](#).

Note

CloudFormation adalah layanan gratis. Namun, Anda dikenakan biaya untuk AWS sumber daya yang Anda sertakan dalam tumpukan Anda dengan tarif saat ini untuk masing-masing. Untuk informasi selengkapnya tentang harga AWS, lihat [halaman detail untuk setiap produk](#).

Topik

- [Gunakan contoh template untuk membuat tumpukan jaringan](#)
- [Gunakan contoh template untuk membuat tumpukan aplikasi web](#)
- [Verifikasi tumpukan berfungsi seperti yang dirancang](#)
- [Memecahkan masalah kesalahan pemetaan AMI](#)
- [Bersihkan sumber daya Anda](#)

Gunakan contoh template untuk membuat tumpukan jaringan

Sebelum Anda memulai panduan ini, periksa apakah Anda memiliki izin IAM untuk menggunakan semua layanan berikut: Amazon VPC, Amazon, dan. EC2 CloudFormation

Tumpukan jaringan berisi VPC, grup keamanan, dan subnet yang akan Anda gunakan dalam tumpukan aplikasi web. Selain sumber daya ini, tumpukan jaringan membuat Gateway Internet dan tabel perutean untuk mengaktifkan akses publik.

Anda harus membuat tumpukan ini sebelum membuat tumpukan aplikasi web. Jika Anda membuat tumpukan aplikasi web terlebih dahulu, tumpukan tidak akan memiliki grup keamanan atau subnet.

Template stack tersedia dari URL berikut: <https://s3.amazonaws.com/cloudformation-examples/user-guide/cross-stack/SampleNetworkCrossStack.template>. Untuk melihat sumber daya yang akan membuat tumpukan, pilih tautan, yang membuka templat. Di Outputs bagian ini, Anda dapat melihat sumber daya jaringan yang diekspor template sampel. Nama sumber daya yang diekspor diawali dengan nama tumpukan jika Anda mengekspor sumber daya jaringan dari tumpukan lainnya. Ketika mengimpor sumber daya jaringan, pengguna dapat menentukan dari tumpukan mana sumber daya akan diimpor.

Untuk membuat tumpukan jaringan

1. Masuk ke AWS Management Console dan buka AWS CloudFormation konsol di <https://console.aws.amazon.com/cloudformation>.
2. Pada halaman Stacks, pilih Buat tumpukan di kanan atas, lalu pilih Dengan sumber daya baru (standar).
3. Pilih Pilih templat yang ada, dan di bagian Tentukan templat, pilih URL Amazon S3.
4. Untuk URL Amazon S3, rekatkan URL berikut: **`https://s3.amazonaws.com/cloudformation-examples/user-guide/cross-stack/SampleNetworkCrossStack.template`**
5. Pilih Berikutnya.
6. Untuk Nama tumpukan, ketik **SampleNetworkCrossStack**, lalu pilih Selanjutnya.

Note

Catat nama tumpukan ini. Anda akan membutuhkan nama tumpukan saat meluncurkan tumpukan aplikasi web.

7. Pilih Selanjutnya. Untuk panduan ini, Anda tidak perlu menambahkan tag atau menentukan pengaturan lanjutan.
8. Pastikan bahwa nama tumpukan dan URL templat sudah benar, lalu pilih Buat tumpukan.

Mungkin perlu beberapa menit CloudFormation untuk membuat tumpukan Anda. Tunggu sampai semua sumber daya telah berhasil dibuat sebelum melanjutkan untuk membuat tumpukan aplikasi web.

9. Untuk memantau kemajuan, tampilkan peristiwa tumpukan. Untuk informasi selengkapnya, lihat [Pantau kemajuan tumpukan](#).

Gunakan contoh template untuk membuat tumpukan aplikasi web

Tumpukan aplikasi web membuat EC2 instance yang menggunakan grup keamanan dan subnet dari tumpukan jaringan.

Anda harus membuat tumpukan ini Wilayah AWS sama dengan tumpukan jaringan.

Template stack tersedia dari URL berikut: <https://s3.amazonaws.com/cloudformation-examples/user-guide/cross-stack/SampleWebAppCrossStack.template>. Untuk melihat sumber daya yang akan membuat tumpukan, pilih tautan, yang akan membuka templat. Di Resources bagian ini, lihat properti EC2 instance. Anda dapat melihat cara sumber daya jaringan diimpor dari tumpukan lain dengan menggunakan fungsi `Fn::ImportValue`.

Untuk membuat tumpukan aplikasi web

1. Dari halaman Stacks, pilih Buat tumpukan di kanan atas, lalu pilih Dengan sumber daya baru (standar).
2. Pilih Pilih templat yang ada, dan di bagian Tentukan templat, pilih URL Amazon S3.
3. Untuk URL Amazon S3, rekatkan URL berikut: **`https://s3.amazonaws.com/cloudformation-examples/user-guide/cross-stack/SampleWebAppCrossStack.template`**
4. Pilih Berikutnya.
5. Untuk Nama tumpukan, ketik **SampleWebAppCrossStack**. Di bagian Parameter, gunakan nilai default untuk NetworkStackNameparameter, lalu pilih Berikutnya.

Templat sampel menggunakan nilai parameter untuk menentukan dari tumpukan mana nilai akan diimpor.

6. Pilih Berikutnya. Untuk panduan ini, Anda tidak perlu menambahkan tag atau menentukan pengaturan lanjutan.
7. Pastikan bahwa nama tumpukan dan URL templat sudah benar, lalu pilih Buat tumpukan.

Mungkin perlu beberapa menit CloudFormation untuk membuat tumpukan Anda.

Verifikasi tumpukan berfungsi seperti yang dirancang

Setelah tumpukan dibuat, lihat sumber daya dan catat ID instans. Untuk informasi selengkapnya tentang melihat sumber daya tumpukan, lihat [Lihat informasi tumpukan dari CloudFormation konsol](#).

Untuk memverifikasi grup keamanan instans dan subnet, lihat properti instans di [EC2 konsol Amazon](#). Jika instans menggunakan grup keamanan dan subnet dari tumpukan `SampleNetworkCrossStack`, Anda telah berhasil membuat referensi lintas tumpukan.

Gunakan konsol tersebut untuk melihat output tumpukan dan contoh URL situs web untuk memverifikasi bahwa aplikasi web berjalan. Untuk informasi selengkapnya, lihat [Lihat informasi tumpukan dari CloudFormation konsol](#).

Memecahkan masalah kesalahan pemetaan AMI

Jika Anda menerima kesalahan `Template error: Unable to get mapping for AWSRegionArch2AMI::[region]::HVM64`, template tidak menyertakan pemetaan AMI untuk Anda Wilayah AWS. Alih-alih memperbarui pemetaan, sebaiknya gunakan parameter publik Systems Manager untuk mereferensikan yang terbaru secara dinamis: AMIs

1. Unduh `SampleWebAppCrossStack` template ke mesin lokal Anda dari: <https://s3.amazonaws.com/cloudformation-examples/user-guide/cross-stack/SampleWebAppCrossStack.template>.
2. Hapus seluruh bagian `AWSRegionArch2AMI` pemetaan.
3. Tambahkan parameter Systems Manager berikut:

```
"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. Ganti `ImageId` referensi yang ada:

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

Dengan:

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

Parameter ini secara otomatis menyelesaikan AMI Amazon Linux 2 terbaru untuk wilayah tempat Anda menyebarkan tumpukan.

Untuk distribusi Linux lainnya, gunakan jalur parameter yang sesuai. Untuk informasi selengkapnya, lihat [Menemukan parameter publik di Parameter Store](#) di Panduan AWS Systems Manager Pengguna.

5. Unggah template yang dimodifikasi ke bucket S3 di akun Anda:

```
aws s3 cp SampleWebAppCrossStack.template s3://amzn-s3-demo-bucket/
```

6. Saat membuat tumpukan, tentukan URL template S3 Anda alih-alih URL contoh.

Bersihkan sumber daya Anda

Untuk memastikan bahwa Anda tidak dikenakan biaya untuk layanan yang tidak diinginkan, hapus tumpukan.

Untuk menghapus tumpukan

1. Di CloudFormation konsol, pilih **SampleWebAppCrossStack** tumpukan.
2. Pilih Tindakan, lalu pilih Hapus tumpukan.
3. Pada pesan konfirmasi, pilih Hapus.
4. Setelah tumpukan dihapus, ulangi langkah yang sama untuk **SampleNetworkCrossStack** tumpukan.

 Note

Tunggu sampai CloudFormation benar-benar menghapus `SampleWebAppCrossStack` tumpukan. Jika EC2 instance masih berjalan di VPC, tidak CloudFormation akan menghapus VPC di tumpukan. `SampleNetworkCrossStack`

Rekan dengan VPC di yang lain Akun AWS

Anda dapat mengintip dengan Virtual Private Cloud (VPC) di tempat lain dengan menggunakan Akun AWS [AWS::EC2::VPCPeeringConnection](#). Ini menciptakan koneksi jaringan antara dua VPCs yang memungkinkan Anda untuk mengarahkan lalu lintas di antara mereka sehingga mereka dapat berkomunikasi seolah-olah mereka berada dalam jaringan yang sama. Koneksi peering VPC dapat membantu memfasilitasi akses data dan transfer data.

Untuk membuat koneksi peering VPC, Anda perlu mengotorisasi dua terpisah Akun AWS dalam satu tumpukan. CloudFormation

Untuk informasi selengkapnya tentang peering VPC dan batasannya, lihat Panduan Peering [VPC](#) Amazon.

Prasyarat

1. Anda memerlukan ID VPC peer, ID Akun AWS rekan, dan peran [akses lintas akun untuk](#) koneksi peering.

 Note

Panduan ini mengacu pada dua akun: Pertama adalah akun yang memungkinkan peering lintas akun (akun penerima). Kedua adalah akun yang meminta koneksi peering (akun peminta).

2. Untuk menerima koneksi peering VPC, peran akses lintas-akun harus dapat Anda tanggung. Sumber daya berperilaku dengan cara yang sama seperti sumber daya koneksi peering VPC di akun yang sama. Untuk informasi tentang cara administrator IAM memberikan izin untuk mengambil peran lintas akun, lihat [Memberikan izin pengguna untuk beralih peran dalam Panduan Pengguna](#) IAM.

Langkah 1: Buat VPC dan peran lintas akun

Di langkah ini, Anda akan membuat VPC dan peran dalam akun penerima.

Untuk membuat VPC dan peran akses lintas akun

1. Masuk ke AWS Management Console dan buka AWS CloudFormation konsol di <https://console.aws.amazon.com/cloudformation>.
2. Dari halaman Stacks, pilih Buat tumpukan di kanan atas, lalu pilih Dengan sumber daya baru (standar).
3. Untuk Prasyarat - Siapkan templat, pilih Pilih templat yang ada lalu Unggah file templat, Pilih file.
4. Buka editor teks di mesin lokal Anda dan tambahkan salah satu templat berikut. Simpan file dan kembali ke konsol untuk memilihnya sebagai file template.

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. Pilih Berikutnya.
6. Beri nama tumpukan (misalnya, **VPC-owner**), lalu masukkan Akun AWS ID akun pemohon di PeerRequesterAccountIdbidang.
7. Terima default, dan kemudian pilih Selanjutnya.
8. Pilih Saya mengakui yang AWS CloudFormation mungkin membuat sumber daya IAM, lalu pilih Buat tumpukan.

Langkah 2: Buat templat yang mencakup **AWS::EC2::VPCPeeringConnection**

Sekarang setelah Anda membuat VPC dan peran lintas akun, Anda dapat mengintip VPC menggunakan yang lain Akun AWS (akun pemohon).

Untuk membuat template yang menyertakan sumber daya [AWS::EC2: VPCPeering Connection](#)

1. Kembali ke halaman beranda AWS CloudFormation konsol.
2. Dari halaman Stacks, pilih Buat tumpukan di kanan atas, lalu pilih Dengan sumber daya baru (standar).
3. Untuk Prasyarat - Siapkan templat, pilih Pilih templat yang ada lalu Unggah file templat, Pilih file.
4. Buka editor teks di mesin lokal Anda dan tambahkan salah satu templat berikut. Simpan file dan kembali ke konsol untuk memilihnya sebagai file template.

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. Pilih Berikutnya.
6. Beri nama tumpukan (misalnya, **VPC-peering-connection**).
7. Terima default, dan kemudian pilih Selanjutnya.
8. Pilih Saya mengakui yang AWS CloudFormation mungkin membuat sumber daya IAM, lalu pilih Buat tumpukan.

Buat template dengan kebijakan yang sangat ketat

Anda mungkin ingin membuat kebijakan yang sangat ketat untuk mengintip VPC Anda dengan yang lain. Akun AWS

Contoh templat berikut menunjukkan bagaimana mengubah templat pemilik peer VPC (akun penerima yang dibuat di Langkah 1 di atas) sehingga lebih ketat.

Example JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "Create a VPC and an assumable role for cross account VPC peering.",
  "Parameters": {
    "PeerRequesterAccountId": {
      "Type": "String"
    }
  },
  "Resources": {
    "peerRole": {
      "Type": "AWS::IAM::Role",
      "Properties": {
        "AssumeRolePolicyDocument": {
          "Statement": [
            {
              "Action": [
                "sts:AssumeRole"
              ],
              "Effect": "Allow",
              "Principal": {
                "AWS": {
                  "Ref": "PeerRequesterAccountId"
                }
              }
            }
          ]
        }
      }
    },
    "Path": "/",
    "Policies": [
      {
        "PolicyDocument": {
          "Statement": [
            {

```

```

        "Action": "ec2:acceptVpcPeeringConnection",
        "Effect": "Allow",
        "Resource": {
            "Fn::Sub": "arn:aws:ec2:${AWS::Region}:${AWS::AccountId}:vpc/${vpc}"
        }
    },
    {
        "Action": "ec2:acceptVpcPeeringConnection",
        "Condition": {
            "StringEquals": {
                "ec2:AcceptorVpc": {
                    "Fn::Sub": "arn:aws:ec2:${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:AccepterVpc':
            'Fn::Sub': 'arn:aws:ec2:${AWS::Region}:${AWS::AccountId}:vpc/
${vpc}'
            Effect: Allow
            Resource:
                'Fn::Sub': >-
                    arn:aws:ec2:${AWS::Region}:${AWS::AccountId}:vpc-peering-
connection/*
            Version: 2012-10-17
            PolicyName: root
    vpc:
        Type: AWS::EC2::VPC
        Properties:
            CidrBlock: 10.1.0.0/16
            EnableDnsHostnames: false
            EnableDnsSupport: false
            InstanceTenancy: default
    Outputs:
        RoleARN:
            Value:
                'Fn::GetAtt':
                    - peerRole
                    - Arn
        VPCId:
            Value:
                Ref: vpc

```

Untuk mengakses VPC, Anda dapat menggunakan templat peminta yang sama seperti pada Langkah 2 di atas.

Untuk informasi selengkapnya, lihat [Identitas dan manajemen akses untuk mengintip VPC di Panduan Peering](#) VPC Amazon.

Buat aplikasi yang diskalakan dan seimbang beban

Untuk panduan ini, Anda membuat tumpukan yang membantu Anda mengatur aplikasi yang diskalakan dan seimbang beban. Panduan ini menyediakan contoh template yang Anda gunakan untuk membuat tumpukan. Contoh template menyediakan grup Auto Scaling, Application Load Balancer, grup keamanan yang mengontrol lalu lintas ke penyeimbang beban dan ke grup Auto Scaling, dan konfigurasi notifikasi Amazon SNS untuk mempublikasikan pemberitahuan tentang aktivitas penskalaan.

Template ini membuat satu atau beberapa EC2 instans Amazon dan Application Load Balancer. Anda akan ditagih untuk sumber daya AWS yang digunakan jika Anda membuat tumpukan dari templat ini.

Template tumpukan penuh

Mari kita mulai dengan template.

YAML

```
AWSTemplateFormatVersion: 2010-09-09
Parameters:
  InstanceType:
    Description: The EC2 instance type
    Type: String
    Default: t3.micro
    AllowedValues:
      - t3.micro
      - t3.small
      - t3.medium
  KeyName:
    Description: Name of an existing EC2 key pair to allow SSH access to the instances
    Type: 'AWS::EC2::KeyPair::KeyName'
  LatestAmiId:
    Description: The latest Amazon Linux 2 AMI from the Parameter Store
    Type: 'AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>'
    Default: '/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2'
  OperatorEmail:
    Description: The email address to notify when there are any scaling activities
    Type: String
  SSHLocation:
    Description: The IP address range that can be used to SSH to the EC2 instances
    Type: String
    MinLength: 9
    MaxLength: 18
    Default: 0.0.0.0/0
    ConstraintDescription: must be a valid IP CIDR range of the form x.x.x.x/x.
  Subnets:
    Type: 'List<AWS::EC2::Subnet::Id>'
    Description: At least two public subnets in different Availability Zones in the
selected VPC
  VPC:
    Type: 'AWS::EC2::VPC::Id'
```

Description: A virtual private cloud (VPC) that enables resources in public subnets to connect to the internet

Resources:

ELBSecurityGroup:

Type: AWS::EC2::SecurityGroup

Properties:

GroupDescription: ELB Security Group

VpcId: !Ref VPC

SecurityGroupIngress:

- IpProtocol: tcp
- FromPort: 80
- ToPort: 80
- CidrIp: 0.0.0.0/0

EC2SecurityGroup:

Type: AWS::EC2::SecurityGroup

Properties:

GroupDescription: EC2 Security Group

VpcId: !Ref VPC

SecurityGroupIngress:

- IpProtocol: tcp
- FromPort: 80
- ToPort: 80
- SourceSecurityGroupId:
 - Fn::GetAtt:
 - ELBSecurityGroup
 - GroupId
- IpProtocol: tcp
- FromPort: 22
- ToPort: 22
- CidrIp: !Ref SSHLocation

EC2TargetGroup:

Type: AWS::ElasticLoadBalancingV2::TargetGroup

Properties:

HealthCheckIntervalSeconds: 30

HealthCheckProtocol: HTTP

HealthCheckTimeoutSeconds: 15

HealthyThresholdCount: 5

Matcher:

HttpCode: '200'

Name: EC2TargetGroup

Port: 80

Protocol: HTTP

TargetGroupAttributes:

- Key: deregistration_delay.timeout_seconds

```
    Value: '20'
    UnhealthyThresholdCount: 3
    VpcId: !Ref VPC
ALBListener:
  Type: AWS::ElasticLoadBalancingV2::Listener
  Properties:
    DefaultActions:
      - Type: forward
        TargetGroupArn: !Ref EC2TargetGroup
    LoadBalancerArn: !Ref ApplicationLoadBalancer
    Port: 80
    Protocol: HTTP
ApplicationLoadBalancer:
  Type: AWS::ElasticLoadBalancingV2::LoadBalancer
  Properties:
    Scheme: internet-facing
    Subnets: !Ref Subnets
    SecurityGroups:
      - !GetAtt ELBSecurityGroup.GroupId
LaunchTemplate:
  Type: AWS::EC2::LaunchTemplate
  Properties:
    LaunchTemplateName: !Sub ${AWS::StackName}-launch-template
    LaunchTemplateData:
      ImageId: !Ref LatestAmiId
      InstanceType: !Ref InstanceType
      KeyName: !Ref KeyName
      SecurityGroupIds:
        - !Ref EC2SecurityGroup
      UserData:
        Fn::Base64: !Sub |
          #!/bin/bash
          yum update -y
          yum install -y httpd
          systemctl start httpd
          systemctl enable httpd
          echo "<h1>Hello World!</h1>" > /var/www/html/index.html
NotificationTopic:
  Type: AWS::SNS::Topic
  Properties:
    Subscription:
      - Endpoint: !Ref OperatorEmail
        Protocol: email
WebServerGroup:
```



```

Type: AWS::AutoScaling::AutoScalingGroup
Properties:
  LaunchTemplate:
    LaunchTemplateId: !Ref LaunchTemplate
    Version: !GetAtt LaunchTemplate.LatestVersionNumber
  MaxSize: '3'
  MinSize: '1'
  NotificationConfigurations:
    - TopicARN: !Ref NotificationTopic
      NotificationTypes: ['autoscaling:EC2_INSTANCE_LAUNCH',
'autoscaling:EC2_INSTANCE_LAUNCH_ERROR', 'autoscaling:EC2_INSTANCE_TERMINATE',
'autoscaling:EC2_INSTANCE_TERMINATE_ERROR']
  TargetGroupARNs:
    - !Ref EC2TargetGroup
  VPCZoneIdentifier: !Ref Subnets

```

JSON

```

{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Parameters": {
    "InstanceType": {
      "Description": "The EC2 instance type",
      "Type": "String",
      "Default": "t3.micro",
      "AllowedValues": [
        "t3.micro",
        "t3.small",
        "t3.medium"
      ]
    },
    "KeyName": {
      "Description": "Name of an existing EC2 key pair to allow SSH access to the instances",
      "Type": "AWS::EC2::KeyPair::KeyName"
    },
    "LatestAmiId": {
      "Description": "The latest Amazon Linux 2 AMI from the Parameter Store",
      "Type": "AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>",
      "Default": "/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2"
    },
    "OperatorEmail": {

```

```
    "Description": "The email address to notify when there are any scaling
activities",
    "Type": "String"
  },
  "SSHLocation": {
    "Description": "The IP address range that can be used to SSH to the EC2
instances",
    "Type": "String",
    "MinLength": 9,
    "MaxLength": 18,
    "Default": "0.0.0.0/0",
    "ConstraintDescription": "Must be a valid IP CIDR range of the form x.x.x.x/x."
  },
  "Subnets": {
    "Type": "List<AWS::EC2::Subnet::Id>",
    "Description": "At least two public subnets in different Availability Zones in the
selected VPC"
  },
  "VPC": {
    "Type": "AWS::EC2::VPC::Id",
    "Description": "A virtual private cloud (VPC) that enables resources in public
subnets to connect to the internet"
  }
},
"Resources": {
  "ELBSecurityGroup": {
    "Type": "AWS::EC2::SecurityGroup",
    "Properties": {
      "GroupDescription": "ELB Security Group",
      "VpcId": {
        "Ref": "VPC"
      }
    },
    "SecurityGroupIngress": [
      {
        "IpProtocol": "tcp",
        "FromPort": 80,
        "ToPort": 80,
        "CidrIp": "0.0.0.0/0"
      }
    ]
  },
  "EC2SecurityGroup": {
    "Type": "AWS::EC2::SecurityGroup",
```

```
"Properties":{
  "GroupDescription":"EC2 Security Group",
  "VpcId":{
    "Ref":"VPC"
  },
  "SecurityGroupIngress":[
    {
      "IpProtocol":"tcp",
      "FromPort":80,
      "ToPort":80,
      "SourceSecurityGroupId":{
        "Fn::GetAtt":[
          "ELBSecurityGroup",
          "GroupId"
        ]
      }
    },
    {
      "IpProtocol":"tcp",
      "FromPort":22,
      "ToPort":22,
      "CidrIp":{
        "Ref":"SSHLocation"
      }
    }
  ]
},
"EC2TargetGroup":{
  "Type":"AWS::ElasticLoadBalancingV2::TargetGroup",
  "Properties":{
    "HealthCheckIntervalSeconds":30,
    "HealthCheckProtocol":"HTTP",
    "HealthCheckTimeoutSeconds":15,
    "HealthyThresholdCount":5,
    "Matcher":{
      "HttpCode":"200"
    },
    "Name":"EC2TargetGroup",
    "Port":80,
    "Protocol":"HTTP",
    "TargetGroupAttributes":[
      {
        "Key":"deregistration_delay.timeout_seconds",
```

```

        "Value": "20"
      }
    ],
    "UnhealthyThresholdCount": 3,
    "VpcId": {
      "Ref": "VPC"
    }
  }
},
"ALBListener": {
  "Type": "AWS::ElasticLoadBalancingV2::Listener",
  "Properties": {
    "DefaultActions": [
      {
        "Type": "forward",
        "TargetGroupArn": {
          "Ref": "EC2TargetGroup"
        }
      }
    ],
    "LoadBalancerArn": {
      "Ref": "ApplicationLoadBalancer"
    },
    "Port": 80,
    "Protocol": "HTTP"
  }
},
"ApplicationLoadBalancer": {
  "Type": "AWS::ElasticLoadBalancingV2::LoadBalancer",
  "Properties": {
    "Scheme": "internet-facing",
    "Subnets": {
      "Ref": "Subnets"
    },
    "SecurityGroups": [
      {
        "Fn::GetAtt": [
          "ELBSecurityGroup",
          "GroupId"
        ]
      }
    ]
  }
},
},
},

```

```

"LaunchTemplate":{
  "Type":"AWS::EC2::LaunchTemplate",
  "Properties":{
    "LaunchTemplateName":{
      "Fn::Sub":"${AWS::StackName}-launch-template"
    },
    "LaunchTemplateData":{
      "ImageId":{
        "Ref":"LatestAmiId"
      },
      "InstanceType":{
        "Ref":"InstanceType"
      },
      "KeyName":{
        "Ref":"KeyName"
      },
      "SecurityGroupIds":[
        {
          "Ref":"EC2SecurityGroup"
        }
      ],
      "UserData":{
        "Fn::Base64":{
          "Fn::Join":[
            "",
            [
              "#!/bin/bash\n",
              "yum update -y\n",
              "yum install -y httpd\n",
              "systemctl start httpd\n",
              "systemctl enable httpd\n",
              "echo \"<h1>Hello World!</h1>\" > /var/www/html/index.html"
            ]
          ]
        }
      }
    }
  },
  "NotificationTopic":{
    "Type":"AWS::SNS::Topic",
    "Properties":{
      "Subscription":[
        {

```

```

        "Endpoint":{
            "Ref":"OperatorEmail"
        },
        "Protocol":"email"
    }
]
},
"WebServerGroup":{
    "Type":"AWS::AutoScaling::AutoScalingGroup",
    "Properties":{
        "LaunchTemplate":{
            "LaunchTemplateId":{
                "Ref":"LaunchTemplate"
            },
            "Version":{
                "Fn::GetAtt":[
                    "LaunchTemplate",
                    "LatestVersionNumber"
                ]
            }
        },
        "MaxSize":"3",
        "MinSize":"1",
        "NotificationConfigurations":[
            {
                "TopicARN":{
                    "Ref":"NotificationTopic"
                },
                "NotificationTypes":[
                    "autoscaling:EC2_INSTANCE_LAUNCH",
                    "autoscaling:EC2_INSTANCE_LAUNCH_ERROR",
                    "autoscaling:EC2_INSTANCE_TERMINATE",
                    "autoscaling:EC2_INSTANCE_TERMINATE_ERROR"
                ]
            }
        ],
        "TargetGroupARNs":[
            {
                "Ref":"EC2TargetGroup"
            }
        ],
        "VPCZoneIdentifier":{
            "Ref":"Subnets"
        }
    }
}

```

```
}  
}  
}  
}  
}
```

Panduan templat

Bagian pertama dari template ini menentukan. `Parameters` Setiap parameter harus diberi nilai saat runtime AWS CloudFormation agar berhasil menyediakan tumpukan. Sumber daya yang ditentukan kemudian dalam templat mereferensikan nilai-nilai ini dan menggunakan data.

- `InstanceType`: Jenis EC2 instance yang disediakan Amazon EC2 Auto Scaling. Jika tidak ditentukan, default dari `t3.micro` digunakan.
- `KeyName`: Sebuah EC2 key pair yang ada untuk memungkinkan akses SSH ke instance.
- `LatestAmiId`: Amazon Machine Image (AMI) untuk instans. Jika tidak ditentukan, instans Anda diluncurkan dengan Amazon Linux 2 AMI, menggunakan parameter AWS Systems Manager publik yang dikelola oleh AWS. Untuk informasi selengkapnya, lihat [Menemukan parameter publik](#) di Panduan AWS Systems Manager Pengguna.
- `OperatorEmail`: Alamat email tempat Anda ingin mengirim pemberitahuan aktivitas penskalaan.
- `SSHLocation`: Rentang alamat IP yang dapat digunakan untuk SSH ke instance.
- `Subnets`: Setidaknya dua subnet publik di Availability Zone yang berbeda.
- `VPC` Virtual Private Cloud (VPC) di akun Anda yang memungkinkan sumber daya di subnet publik untuk terhubung ke internet.

Note

Anda dapat menggunakan VPC default dan subnet default untuk memungkinkan instance mengakses internet. Jika menggunakan VPC Anda sendiri, pastikan bahwa itu memiliki subnet yang dipetakan ke setiap Availability Zone dari Wilayah tempat Anda bekerja. Minimal, Anda harus memiliki dua subnet publik yang tersedia untuk membuat penyeimbang beban.

Bagian selanjutnya dari template ini menentukan. `Resources` Bagian ini menentukan sumber daya tumpukan dan propertinya.

[AWS::EC2::SecurityGroup](#)sumber daya ELBSecurityGroup

- SecurityGroupIngressberisi aturan masuknya TCP yang memungkinkan akses dari semua alamat IP (" CidrIp ": "0.0.0.0/0") pada port 80.

[AWS::EC2::SecurityGroup](#)sumber daya EC2SecurityGroup

- SecurityGroupIngressberisi dua aturan ingress: 1) aturan masuknya TCP yang memungkinkan akses SSH (port 22) dari rentang alamat IP yang Anda berikan untuk parameter SSHLocation input dan 2) aturan masuknya TCP yang memungkinkan akses dari penyeimbang beban dengan menentukan grup keamanan penyeimbang beban. [GetAtt](#)Fungsi ini digunakan untuk mendapatkan ID grup keamanan dengan nama logisELBSecurityGroup.

[AWS::ElasticLoadBalancingV2::TargetGroup](#)sumber daya EC2TargetGroup

- Port,Protocol, dan HealthCheckProtocol tentukan port EC2 instance (80) dan protokol (HTTP) yang digunakan lalu lintas ApplicationLoadBalancer rute dan Elastic Load Balancing yang digunakan untuk memeriksa kesehatan instance. EC2
- HealthCheckIntervalSecondsmenentukan bahwa EC2 instance memiliki interval 30 detik antara pemeriksaan kesehatan. HealthCheckTimeoutSecondsini didefinisikan sebagai lamanya waktu Elastic Load Balancing menunggu respons dari target pemeriksaan kesehatan (15 detik dalam contoh ini). Setelah periode batas waktu berlalu, Elastic Load Balancing menandai pemeriksaan kesehatan instans EC2 tersebut sebagai tidak sehat. Ketika sebuah EC2 instance gagal dalam tiga pemeriksaan kesehatan berturut-turut (UnhealthyThresholdCount), Elastic Load Balancing menghentikan perutean lalu lintas ke instance tersebut hingga instance EC2 tersebut memiliki lima pemeriksaan kesehatan sehat berturut-turut (). HealthyThresholdCount Pada saat itu, Elastic Load Balancing menganggap instance sehat dan mulai merutekan lalu lintas ke instance lagi.
- TargetGroupAttributesmemperbarui nilai penundaan deregistrasi grup target menjadi 20 detik. Secara default, Elastic Load Balancing menunggu 300 detik sebelum menyelesaikan proses deregistrasi.

[AWS::ElasticLoadBalancingV2::Listener](#)sumber daya ALBListener

- `DefaultActions` menentukan port yang didengarkan penyeimbang beban, grup target tempat penyeimbang beban meneruskan permintaan, dan protokol yang digunakan untuk merutekan permintaan.

[`AWS::ElasticLoadBalancingV2::LoadBalancer`](#) sumber daya `ApplicationLoadBalancer`

- `Subnets` mengambil nilai parameter `Subnets` input sebagai daftar subnet publik di mana node penyeimbang beban akan dibuat.
- `SecurityGroup` mendapatkan ID grup keamanan yang bertindak sebagai firewall virtual untuk node penyeimbang beban Anda untuk mengontrol lalu lintas masuk. [`GetAtt`](#) Fungsi ini digunakan untuk mendapatkan ID grup keamanan dengan nama logis `ELBSecurityGroup`.

[`AWS::EC2::LaunchTemplate`](#) sumber daya `LaunchTemplate`

- `ImageId` mengambil nilai parameter `LatestAmiId` input sebagai AMI untuk digunakan.
- `KeyName` mengambil nilai parameter `KeyName` input sebagai EC2 key pair yang akan digunakan.
- `SecurityGroupIds` mendapatkan ID grup keamanan dengan nama logis `EC2SecurityGroup` yang bertindak sebagai firewall virtual untuk EC2 instance Anda untuk mengontrol lalu lintas masuk.
- `UserData` adalah skrip konfigurasi yang berjalan setelah instance aktif dan berjalan. Dalam contoh ini, skrip menginstal Apache dan membuat file `index.html`.

[`AWS::SNS::Topic`](#) sumber daya `NotificationTopic`

- `Subscription` mengambil nilai parameter `OperatorEmail` input sebagai alamat email untuk penerima notifikasi ketika ada aktivitas penskalaan.

`AWS::AutoScaling::AutoScaling` Sumber daya [`grup`](#) `WebServerGroup`

- `MinSize` dan `MaxSize` atur jumlah EC2 instans minimum dan maksimum dalam grup `Auto Scaling`.
- `TargetGroupARNs` mengambil ARN dari kelompok target dengan nama logis. `EC2TargetGroup` Saat grup `Auto Scaling` ini menskalakan, grup `Auto Scaling` secara otomatis mendaftarkan dan membatalkan pendaftaran instance dengan grup target ini.

- `VPCZoneIdentifier` mengambil nilai parameter `Subnets` input sebagai daftar subnet publik tempat EC2 instance dapat dibuat.

Langkah 1: Luncurkan tumpukan

Sebelum Anda meluncurkan tumpukan, periksa apakah Anda memiliki izin AWS Identity and Access Management (IAM) untuk menggunakan semua layanan berikut: Amazon, EC2, Amazon Auto Scaling, AWS Systems Manager, Elastic Load Balancing, Amazon SNS, dan AWS CloudFormation.

Prosedur berikut melibatkan mengunggah template tumpukan sampel dari file. Buka editor teks di mesin lokal Anda dan tambahkan salah satu templat. Simpan file dengan nama `sampleloadbalancedappstack.template`.

Untuk meluncurkan template tumpukan

1. Masuk ke AWS Management Console dan buka AWS CloudFormation konsol di <https://console.aws.amazon.com/cloudformation>.
2. Pilih Buat tumpukan, Dengan sumber daya baru (standar).
3. Di bawah Tentukan templat, pilih Unggah file templat, Pilih file untuk mengunggah `sampleloadbalancedappstack.template` file.
4. Pilih Berikutnya.
5. Pada halaman Tentukan detail tumpukan, ketik nama tumpukan (misalnya, **SampleLoadBalancedAppStack**).
6. Di bawah Parameter, tinjau parameter untuk tumpukan dan berikan nilai untuk semua parameter yang tidak memiliki nilai default, termasuk `OperatorEmail`, `SSHLocationKeyName`, `VPC`, dan `Subnet`.
7. Pilih Selanjutnya dua kali.
8. Pada halaman Ulasan, tinjau dan konfirmasi pengaturan.
9. Pilih Kirim.

Anda dapat melihat status tumpukan di AWS CloudFormation konsol di kolom Status. Ketika AWS CloudFormation telah berhasil membuat tumpukan, Anda menerima status `CREATE_COMPLETE`.

 Note

Setelah Anda membuat tumpukan, Anda harus mengonfirmasi langganan sebelum alamat email dapat mulai menerima pemberitahuan. Untuk informasi selengkapnya, lihat [Mendapatkan notifikasi Amazon SNS saat grup Auto Scaling Anda menskalakan di Panduan](#) Pengguna Amazon Auto EC2 Scaling.

Langkah 2: Bersihkan sumber daya sampel Anda

Untuk memastikan bahwa Anda tidak dikenakan biaya untuk sumber daya sampel yang tidak digunakan, hapus tumpukan.

Untuk menghapus tumpukan

1. Di AWS CloudFormation konsol, pilih SampleLoadBalancedAppStacktumpukan.
2. Pilih Hapus.
3. Dalam pesan konfirmasi, pilih Hapus tumpukan.

Status SampleLoadBalancedAppStackperubahan pada DELETE_IN_PROGRESS. Ketika AWS CloudFormation menyelesaikan penghapusan tumpukan, itu menghapus tumpukan dari daftar.

Gunakan contoh template dari panduan ini untuk membangun template tumpukan Anda sendiri. Untuk informasi selengkapnya, lihat [Tutorial: Menyiapkan aplikasi yang diskalakan dan diseimbangkan beban di Panduan](#) Pengguna Amazon EC2 Auto Scaling.

Menyebarkan aplikasi di Amazon EC2

Anda dapat menggunakan CloudFormation untuk menginstal, mengonfigurasi, dan memulai aplikasi secara otomatis di EC2 instans Amazon. Melakukan hal ini memungkinkan Anda untuk dengan mudah menduplikasi penyebaran dan memperbarui instalasi yang ada tanpa terhubung langsung ke instans, yang dapat menghemat banyak waktu dan usaha.

CloudFormation termasuk satu set skrip pembantu (`cfn-init`, `cfn-signal`, `cfn-get-metadata`, dan `cfn-hup`) yang didasarkan pada `cloud-init`. Anda memanggil skrip pembantu ini dari CloudFormation templat Anda untuk menginstal, mengonfigurasi, dan memperbarui aplikasi di EC2

instans Amazon yang ada di templat yang sama. Untuk informasi selengkapnya, lihat [referensi skrip CloudFormation pembantu di Panduan Referensi AWS CloudFormation](#) Template.

Dalam [tutorial memulai](#), Anda membuat server web sederhana menggunakan UserData skrip bash dasar. Meskipun ini berfungsi untuk halaman “Hello World” sederhana, aplikasi nyata seringkali membutuhkan konfigurasi yang lebih canggih, termasuk:

- Beberapa paket perangkat lunak diinstal dalam urutan yang benar.
- File konfigurasi kompleks yang dibuat dengan konten tertentu.
- Layanan dimulai dan dikonfigurasi untuk berjalan secara otomatis.
- Penanganan kesalahan dan validasi proses penyiapan.

CloudFormation skrip pembantu menyediakan cara yang lebih kuat dan dapat dipelihara untuk mengonfigurasi EC2 instance dibandingkan dengan skrip bash dasar di. UserData Skrip cfn-init helper membaca data konfigurasi dari metadata template Anda dan menerapkannya secara sistematis ke instance Anda.

Dalam tutorial ini, Anda akan belajar cara menggunakan skrip cfn-init helper dan memantau proses bootstrap.

Note

CloudFormation gratis, tetapi Anda akan dikenakan biaya untuk EC2 sumber daya Amazon yang Anda buat. Namun, jika Anda baru mengenal AWS, Anda dapat memanfaatkan [Tingkat Gratis](#) untuk meminimalkan atau menghilangkan biaya selama proses pembelajaran ini.

Topik

- [Prasyarat](#)
- [Memahami konsep bootstrap](#)
- [Mulailah dengan contoh bootstrap sederhana](#)
- [Menambahkan file dan perintah](#)
- [Menambahkan keamanan jaringan](#)
- [Template bootstrap lengkap](#)
- [Buat tumpukan menggunakan konsol](#)
- [Pantau proses bootstrap](#)

- [Uji server web bootstrapped](#)
- [Memecahkan masalah bootstrap](#)
- [Pembersihan sumber daya](#)
- [Langkah selanjutnya](#)

Prasyarat

- Anda harus telah menyelesaikan [Membuat tumpukan pertama Anda](#) tutorial atau memiliki pengalaman yang setara dengan CloudFormation dasar-dasar.
- Anda harus memiliki akses ke pengguna atau peran IAM yang memiliki izin untuk menggunakan Amazon EC2 dan CloudFormation, atau akses pengguna administratif. Akun AWS
- Anda harus memiliki Virtual Private Cloud (VPC) yang memiliki akses ke internet. Template tutorial ini membutuhkan VPC default, yang datang secara otomatis dengan yang lebih baru. Akun AWS
Jika Anda tidak memiliki VPC default, atau jika telah dihapus, lihat bagian pemecahan masalah di [Membuat tumpukan pertama Anda](#) tutorial untuk solusi alternatif.

Memahami konsep bootstrap

Mari kita pahami konsep kunci yang membuat bootstrapping berfungsi sebelum membuat template.

cfn-initNaskah pembantu

CloudFormation menyediakan skrip pembantu Python yang dapat Anda gunakan untuk menginstal perangkat lunak dan memulai layanan pada instance Amazon. EC2 `cfn-init` Skrip membaca metadata sumber daya dari template Anda dan menerapkan konfigurasi ke instance Anda.

Prosesnya bekerja sebagai berikut:

1. Anda menentukan konfigurasi di Metadata bagian sumber EC2 daya Anda.
2. Anda menelepon `cfn-init` dari UserData naskah.
3. `cfn-init` membaca metadata dan menerapkan konfigurasi.
4. Instans Anda dikonfigurasi sesuai dengan spesifikasi Anda.

Struktur metadata

Konfigurasi didefinisikan dalam struktur tertentu dalam EC2 instance Anda.

```
Resources:
  EC2Instance:
    Type: AWS::EC2::Instance
    Metadata:
      AWS::CloudFormation::Init:
        config:
          packages:
            files:
              commands:
                services:
```

Metadata section for the resource
Required key that cfn-init looks for
Configuration name (you can have multiple)
Install packages
Create files
Run commands
Start/stop services

cfn-initSkrp memproses bagian-bagian ini dalam urutan tertentu: paket, grup, pengguna, sumber, file, perintah, dan kemudian layanan.

Mulailah dengan contoh bootstrap sederhana

Mari kita mulai dengan contoh bootstrap minimal yang hanya menginstal dan memulai Apache.

```
Resources:
  EC2Instance:
    Type: AWS::EC2::Instance
    Metadata:
      AWS::CloudFormation::Init:
        config:
          packages:
            yum:
              httpd: []
          services:
            sysvinit:
              httpd:
                enabled: true
                ensureRunning: true
    Properties:
      ImageId: !Ref LatestAmiId
      InstanceType: !Ref InstanceType
      UserData: !Base64
        Fn::Sub: |
          #!/bin/bash
          yum install -y aws-cfn-bootstrap
          /opt/aws/bin/cfn-init -v --stack ${AWS::StackName} --resource EC2Instance --
region ${AWS::Region}
```

Contoh sederhana ini menunjukkan konsep inti:

- `packages` bagian menginstal `httpd` paket menggunakan `yum`. Ini berfungsi di Amazon Linux dan distribusi Linux lainnya yang digunakannya.
- `services` bagian memastikan `httpd` mulai dan berjalan secara otomatis.
- `UserData` menginstal alat `bootstrap` terbaru dan memanggil `cfn-init`.

Menambahkan file dan perintah

Sekarang, mari kita tingkatkan contoh kita dengan menambahkan halaman web khusus dan file log di `/var/log` direktori pada EC2 instance.

Membuat file

`files` Bagian ini memungkinkan Anda untuk membuat file pada instance dengan konten tertentu. Pipa vertikal (`|`) memungkinkan Anda untuk melewati blok teks literal (kode HTML) sebagai konten file (`/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>
```

Menjalankan perintah

`commands` Bagian ini memungkinkan Anda menjalankan perintah shell selama proses bootstrap. Perintah ini membuat file log di `/var/log/welcome.txt` pada EC2 instance. Untuk melihatnya, Anda memerlukan Amazon EC2 key pair untuk digunakan untuk akses SSH dan rentang alamat IP yang dapat digunakan untuk SSH ke instance (tidak dibahas di sini).

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

Menambahkan keamanan jaringan

Karena kita sedang menyiapkan server web, kita perlu mengizinkan lalu lintas web (HTTP) untuk mencapai EC2 instance kita. Untuk melakukan ini, kami akan membuat grup keamanan yang memungkinkan lalu lintas masuk pada port 80 dari alamat IP Anda. EC2 instance juga perlu mengirim lalu lintas ke internet, misalnya, untuk menginstal pembaruan paket. Secara default, grup keamanan mengizinkan semua lalu lintas keluar. Kami kemudian akan mengaitkan grup keamanan ini dengan EC2 instance kami menggunakan `SecurityGroupIds` properti.

```
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
```

Template bootstrap lengkap

Sekarang, mari kita satukan semua bagian. Berikut adalah template lengkap yang menggabungkan semua konsep yang telah kita bahas.

```
AWSTemplateFormatVersion: 2010-09-09
Description: Bootstrap an EC2 instance with Apache web server using cfn-init

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

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


MyIP:

Description: Your IP address in CIDR format (e.g. 203.0.113.1/32)

Type: String

MinLength: 9

MaxLength: 18

Default: 0.0.0.0/0

AllowedPattern: '^(\d{1,3}\.){3}\d{1,3}\.\/\d{1,2}\$'

ConstraintDescription: must be a valid IP CIDR range of the form x.x.x.x/x.

Resources:**WebServerSecurityGroup:**

Type: AWS::EC2::SecurityGroup

Properties:

GroupDescription: Allow HTTP access from my IP address

SecurityGroupIngress:

- IpProtocol: tcp

Description: HTTP

FromPort: 80

ToPort: 80

CidrIp: !Ref MyIP

WebServer:

Type: AWS::EC2::Instance

Metadata:**AWS::CloudFormation::Init:****config:****packages:****yum:**

httpd: []

files:

/var/www/html/index.html:

content: |

<body>

<h1>Congratulations, you have successfully launched the AWS

CloudFormation sample.</h1>

</body>

commands:**createWelcomeLog:**

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

services:**sysvinit:****httpd:**

enabled: true

```
        ensureRunning: true
    Properties:
        ImageId: !Ref LatestAmiId
        InstanceType: !Ref InstanceType
        SecurityGroupIds:
            - !Ref WebServerSecurityGroup
        UserData: !Base64
            Fn::Sub: |
                #!/bin/bash
                yum install -y aws-cfn-bootstrap
                /opt/aws/bin/cfn-init -v --stack ${AWS::StackName} --resource WebServer --
region ${AWS::Region}
        Tags:
            - Key: Name
              Value: Bootstrap Tutorial Web Server

    Outputs:
        WebsiteURL:
            Value: !Sub 'http://${WebServer.PublicDnsName}'
            Description: EC2 instance public DNS name
```

Buat tumpukan menggunakan konsol

Prosedur berikut melibatkan mengunggah template tumpukan sampel dari file.

Buka editor teks di mesin lokal Anda dan tambahkan template. Simpan file dengan `namasamplelinux2stack.template`.

Untuk meluncurkan template tumpukan

1. Masuk ke AWS Management Console dan buka AWS CloudFormation konsol di <https://console.aws.amazon.com/cloudformation>.
2. Pilih Buat tumpukan, Dengan sumber daya baru (standar).
3. Di bawah Tentukan templat, pilih Unggah file templat, Pilih file untuk mengunggah `samplelinux2stack.template` file.
4. Pilih Berikutnya.
5. Pada halaman Tentukan detail tumpukan, ketik **BootstrapTutorialStack** sebagai nama tumpukan.
6. Di bawah Parameter, lakukan hal berikut.
 - LatestAmild: Tinggalkan nilai default.

- `InstanceType`: Pilih salah satu `t2.micro` atau `t3.micro` untuk jenis instance. EC2
- `MyIP`: Masukkan alamat IP publik Anda dengan akhiran. `/32`

7. Pilih Berikutnya dua kali, lalu Kirim untuk membuat tumpukan.

Pantau proses bootstrap

Proses bootstrap membutuhkan waktu lebih lama daripada EC2 peluncuran sederhana karena perangkat lunak tambahan sedang diinstal dan dikonfigurasi.

Untuk memantau kemajuan bootstrap

1. Di CloudFormation konsol, pilih tumpukan Anda dan buka tab Acara.
2. Perhatikan `WebServer CREATE_IN_PROGRESS` acara tersebut. Proses bootstrap dimulai setelah instance diluncurkan.
3. Proses bootstrap biasanya memakan waktu beberapa menit. Anda akan melihat `WebServer CREATE_COMPLETE` ketika itu selesai.

Jika Anda ingin melihat apa yang terjadi selama proses bootstrap, Anda dapat memeriksa log instance.

Untuk melihat log bootstrap (opsional)

1. Buka [EC2 konsol](#) dan temukan instance Anda.
2. Pilih instance, lalu pilih Actions, Monitor dan troubleshoot, Dapatkan log sistem untuk melihat kemajuan bootstrap.
3. Jika Anda tidak segera melihat log, tunggu dan segarkan halaman.

Uji server web bootstrapped

Saat tumpukan Anda ditampilkan `CREATE_COMPLETE`, uji server web Anda.

Untuk menguji server web

1. Di CloudFormation konsol, buka tab Output untuk tumpukan Anda.
2. Klik pada nilai `WebsiteUrl` untuk membuka server web Anda di tab baru.

3. Anda akan melihat halaman web kustom Anda dengan pesan `Congratulations, you have successfully launched the AWS CloudFormation sample.`

Note

Jika halaman tidak segera dimuat, tunggu sebentar dan coba lagi. Proses bootstrap mungkin masih selesai bahkan setelah tumpukan ditampilkan `CREATE_COMPLETE`.

Memecahkan masalah bootstrap

Jika proses bootstrap Anda gagal atau server web Anda tidak berfungsi, berikut adalah masalah dan solusi umum.

Masalah umum

- Pembuatan tumpukan gagal - Periksa tab Acara untuk pesan kesalahan tertentu.
- Server web tidak dapat diakses — Verifikasi alamat IP Anda benar dalam MyIP parameter. Ingatlah untuk menyertakan `/32` di akhir.
- Proses Bootstrap gagal - Instance mungkin diluncurkan tetapi `cfn-init` gagal. Periksa log sistem seperti yang dijelaskan di bagian pemantauan.

Pembersihan sumber daya

Untuk menghindari tagihan yang sedang berlangsung, Anda dapat membersihkan dengan menghapus tumpukan dan sumber dayanya.

Untuk menghapus tumpukan dan sumber dayanya

1. Buka [konsol CloudFormation](#).
2. Pada halaman Stacks, pilih opsi di sebelah nama tumpukan yang Anda buat (**BootstrapTutorialStack**) dan kemudian pilih Hapus.
3. Saat diminta konfirmasi, pilih Hapus.
4. Pantau kemajuan proses penghapusan tumpukan pada tab Event. Status untuk **BootstrapTutorialStack** perubahan ke `DELETE_IN_PROGRESS`. Ketika CloudFormation menyelesaikan penghapusan tumpukan, itu menghapus tumpukan dari daftar.

Langkah selanjutnya

Selamat! Anda telah berhasil mempelajari cara mem-bootstrap EC2 instance dengan CloudFormation. Anda sekarang mengerti:

- Cara menggunakan `cfn-init` skrip pembantu
- Cara menyusun metadata untuk bootstrap
- Cara menginstal paket, membuat file, menjalankan perintah, dan mengelola layanan
- Cara memantau masalah bootstrap

Untuk terus belajar:

- Pelajari cara mem-bootstrap Windows tumpukan. Untuk informasi selengkapnya, lihat [Tumpukan berbasis bootstrap Windows CloudFormation](#).
- Jelajahi skenario bootstrap yang lebih kompleks dengan beberapa set konfigurasi. Untuk informasi selengkapnya, lihat [cfn-init](#) dan [AWS::CloudFormation::Init](#) di Panduan Referensi AWS CloudFormation Template.
- Pelajari tentang `cfn-signal` pelaporan status penyelesaian bootstrap. Untuk informasi selengkapnya, lihat [cfn-signal di Panduan Referensi AWS CloudFormation Template](#).

Memperbarui tumpukan

Dengan CloudFormation, Anda dapat memperbarui properti untuk sumber daya di tumpukan yang ada. Perubahan ini dapat berkisar dari perubahan konfigurasi sederhana, seperti memperbarui ambang alarm pada CloudWatch alarm, hingga perubahan yang lebih kompleks, seperti memperbarui Amazon Machine Image (AMI) yang berjalan pada EC2 instans Amazon. Banyak sumber AWS daya dalam template dapat diperbarui, dan kami terus menambahkan dukungan untuk lebih banyak.

Bagian ini membahas perkembangan sederhana dari pembaruan tumpukan yang sedang berjalan. Ini menunjukkan bagaimana penggunaan template memungkinkan untuk menggunakan sistem kontrol versi untuk konfigurasi AWS infrastruktur Anda, sama seperti Anda menggunakan kontrol versi untuk perangkat lunak yang Anda jalankan. Kita akan mengikuti panduan dengan langkah-langkah berikut:

1. [Membuat tumpukan awal](#)— membuat tumpukan menggunakan basis Amazon Linux AMI, menginstal Apache Web Server dan aplikasi PHP sederhana menggunakan skrip CloudFormation pembantu.

2. [Memperbarui aplikasi](#)— perbarui salah satu file dalam aplikasi dan gunakan perangkat lunak menggunakan CloudFormation.
3. [Memperbarui tipe instans](#)— ubah jenis instance dari instans Amazon EC2 yang mendasarinya.
4. [Perbarui AMI pada EC2 instans Amazon](#)— ubah Amazon Machine Image (AMI) untuk EC2 instans Amazon di tumpukan Anda.
5. [Menambahkan pasangan kunci ke sebuah instans](#)— tambahkan EC2 key pair Amazon ke instance, lalu perbarui grup keamanan untuk memungkinkan akses SSH ke instance.
6. [Mengubah sumber daya tumpukan](#)— menambah dan menghapus sumber daya dari tumpukan, mengubahnya menjadi aplikasi yang diskalakan otomatis dan seimbang beban dengan memperbarui template.

Sebuah aplikasi sederhana

Kita akan mulai dengan membuat tumpukan yang dapat kita gunakan di seluruh bagian ini. Kami telah menyediakan template sederhana yang meluncurkan satu contoh aplikasi web PHP yang dihosting di Apache Web Server dan berjalan di Amazon Linux AMI.

Apache Web Server, PHP, dan aplikasi PHP sederhana semuanya diinstal oleh skrip CloudFormation pembantu yang diinstal secara default di Amazon Linux AMI. Cuplikan template berikut menunjukkan metadata yang menjelaskan paket dan file yang akan diinstal, dalam hal ini Apache Web Server dan infrastruktur PHP dari repositori Yum untuk Amazon Linux AMI. Snippet juga menunjukkan bagian Layanan, yang memastikan bahwa Apache Web Server berjalan. Di bagian Properties dari definisi EC2 instance Amazon, UserData properti berisi CloudInit skrip yang memanggil cfn-init untuk menginstal paket dan file.

```
"WebServerInstance": {
  "Type" : "AWS::EC2::Instance",
  "Metadata" : {
    "AWS::CloudFormation::Init" : {
      "config" : {
        "packages" : {
          "yum" : {
            "httpd"      : [],
            "php"        : []
          }
        },
        "files" : {
```

```

    "/var/www/html/index.php" : {
      "content" : { "Fn::Join" : ["", [
        "<?php\n",
        "echo '<h1>AWS CloudFormation sample PHP application</h1>';\n",
        "echo '<p>', { "Ref" : "WelcomeMessage" }, "</p>";\n",
        "?>\n"
      ]]],
      "mode"      : "000644",
      "owner"     : "apache",
      "group"     : "apache"
    },
  },

  :

  "services" : {
    "sysvinit" : {
      "httpd"   : { "enabled" : "true", "ensureRunning" : "true" }
    }
  }
}
},
},

"Properties": {
  :
  "UserData" : { "Fn::Base64" : { "Fn::Join" : ["", [
    "#!/bin/bash\n",
    "yum install -y aws-cfn-bootstrap\n",

    :

    "# Install the files and packages from the metadata\n",
    "/opt/aws/bin/cfn-init -v ",
    "    --stack ", { "Ref" : "AWS::StackName" },
    "    --resource WebServerInstance ",
    "    --region ", { "Ref" : "AWS::Region" }, "\n",
    :
  ]]]}
}
},
},

```

Aplikasi itu sendiri adalah contoh dua baris “Hello World” yang sepenuhnya didefinisikan dalam template. Untuk aplikasi dunia nyata, file dapat disimpan di Amazon S3 GitHub,, atau repositori lain dan direferensikan dari template. CloudFormation dapat mengunduh paket (seperti RPMs atau RubyGems), dan referensi file individual dan memperluas .zip dan .tar file untuk membuat artefak aplikasi pada EC2 instance Amazon.

Template mengaktifkan dan mengonfigurasi daemon cfn-hup untuk mendengarkan perubahan pada konfigurasi yang ditentukan dalam metadata untuk instance Amazon. EC2 Dengan menggunakan daemon cfn-hup, Anda dapat memperbarui perangkat lunak aplikasi, seperti versi Apache atau PHP, atau Anda dapat memperbarui file aplikasi PHP itu sendiri. CloudFormation Cuplikan berikut dari EC2 sumber daya Amazon yang sama di template menunjukkan bagian yang diperlukan untuk mengonfigurasi cfn-hup untuk memanggil cfn-init untuk memperbarui perangkat lunak jika ada perubahan pada metadata yang terdeteksi:

```
"WebServerInstance": {
  "Type" : "AWS::EC2::Instance",
  "Metadata" : {
    "AWS::CloudFormation::Init" : {
      "config" : {

        :

        "files" : {

          :

          "/etc/cfn/cfn-hup.conf" : {
            "content" : { "Fn::Join" : [ "", [
              "[main]\n",
              "stack=", { "Ref" : "AWS::StackName" }, "\n",
              "region=", { "Ref" : "AWS::Region" }, "\n"
            ] ] },
            "mode"      : "000400",
            "owner"     : "root",
            "group"     : "root"
          },

          "/etc/cfn/hooks.d/cfn-auto-reloader.conf" : {
            "content": { "Fn::Join" : [ "", [
              "[cfn-auto-reloader-hook]\n",
              "triggers=post.update\n",
```



```

        "path=Resources.WebServerInstance.Metadata.AWS::CloudFormation::Init
\n",
        "action=/opt/aws/bin/cfn-init -s ", { "Ref" : "AWS::StackId" }, " -r
WebServerInstance ",
        " --region      ", { "Ref" : "AWS::Region" }, "\n",
        "runas=root\n"
    ]}]
    }
  },
  :
},

"Properties": {

  :

  "UserData"      : { "Fn::Base64" : { "Fn::Join" : [""], [

    :

    "# Start up the cfn-hup daemon to listen for changes to the Web Server metadata
\n",
    "/opt/aws/bin/cfn-hup || error_exit 'Failed to start cfn-hup'\n",

    :
  ]]]}
  }
},

```

Untuk menyelesaikan tumpukan, template membuat grup EC2 keamanan Amazon.

```

{
  "AWSTemplateFormatVersion" : "2010-09-09",

  "Description" : "AWS CloudFormation Sample Template: Sample template that can be used
to test EC2 updates. **WARNING** This template creates an Amazon Ec2 Instance. You
will be billed for the AWS resources used if you create a stack from this template.",

  "Parameters" : {

    "InstanceType" : {
      "Description" : "WebServer EC2 instance type",
      "Type" : "String",

```

```
"Default" : "t2.small",
"AllowedValues" : [
    "t1.micro",
    "t2.nano",
    "t2.micro",
    "t2.small",
    "t2.medium",
    "t2.large",
    "m1.small",
    "m1.medium",
    "m1.large",
    "m1.xlarge",
    "m2.xlarge",
    "m2.2xlarge",
    "m2.4xlarge",
    "m3.medium",
    "m3.large",
    "m3.xlarge",
    "m3.2xlarge",
    "m4.large",
    "m4.xlarge",
    "m4.2xlarge",
    "m4.4xlarge",
    "m4.10xlarge",
    "c1.medium",
    "c1.xlarge",
    "c3.large",
    "c3.xlarge",
    "c3.2xlarge",
    "c3.4xlarge",
    "c3.8xlarge",
    "c4.large",
    "c4.xlarge",
    "c4.2xlarge",
    "c4.4xlarge",
    "c4.8xlarge",
    "g2.2xlarge",
    "g2.8xlarge",
    "r3.large",
    "r3.xlarge",
    "r3.2xlarge",
    "r3.4xlarge",
    "r3.8xlarge",
    "i2.xlarge",
```

```

        "i2.2xlarge",
        "i2.4xlarge",
        "i2.8xlarge",
        "d2.xlarge",
        "d2.2xlarge",
        "d2.4xlarge",
        "d2.8xlarge",
        "hi1.4xlarge",
        "hs1.8xlarge",
        "cr1.8xlarge",
        "cc2.8xlarge",
        "cg1.4xlarge"
    ],
    "ConstraintDescription" : "must be a valid EC2 instance type."
}
},

"Mappings" : {
    "AWSInstanceType2Arch" : {
        "t1.micro"      : { "Arch" : "HVM64" },
        "t2.nano"      : { "Arch" : "HVM64" },
        "t2.micro"      : { "Arch" : "HVM64" },
        "t2.small"     : { "Arch" : "HVM64" },
        "t2.medium"    : { "Arch" : "HVM64" },
        "t2.large"     : { "Arch" : "HVM64" },
        "m1.small"     : { "Arch" : "HVM64" },
        "m1.medium"    : { "Arch" : "HVM64" },
        "m1.large"     : { "Arch" : "HVM64" },
        "m1.xlarge"    : { "Arch" : "HVM64" },
        "m2.xlarge"    : { "Arch" : "HVM64" },
        "m2.2xlarge"   : { "Arch" : "HVM64" },
        "m2.4xlarge"   : { "Arch" : "HVM64" },
        "m3.medium"    : { "Arch" : "HVM64" },
        "m3.large"     : { "Arch" : "HVM64" },
        "m3.xlarge"    : { "Arch" : "HVM64" },
        "m3.2xlarge"   : { "Arch" : "HVM64" },
        "m4.large"     : { "Arch" : "HVM64" },
        "m4.xlarge"    : { "Arch" : "HVM64" },
        "m4.2xlarge"   : { "Arch" : "HVM64" },
        "m4.4xlarge"   : { "Arch" : "HVM64" },
        "m4.10xlarge"  : { "Arch" : "HVM64" },
        "c1.medium"    : { "Arch" : "HVM64" },
        "c1.xlarge"    : { "Arch" : "HVM64" },
        "c3.large"     : { "Arch" : "HVM64" },

```

```

    "c3.xlarge" : { "Arch" : "HVM64" },
    "c3.2xlarge" : { "Arch" : "HVM64" },
    "c3.4xlarge" : { "Arch" : "HVM64" },
    "c3.8xlarge" : { "Arch" : "HVM64" },
    "c4.large" : { "Arch" : "HVM64" },
    "c4.xlarge" : { "Arch" : "HVM64" },
    "c4.2xlarge" : { "Arch" : "HVM64" },
    "c4.4xlarge" : { "Arch" : "HVM64" },
    "c4.8xlarge" : { "Arch" : "HVM64" },
    "g2.2xlarge" : { "Arch" : "HVMG2" },
    "g2.8xlarge" : { "Arch" : "HVMG2" },
    "r3.large" : { "Arch" : "HVM64" },
    "r3.xlarge" : { "Arch" : "HVM64" },
    "r3.2xlarge" : { "Arch" : "HVM64" },
    "r3.4xlarge" : { "Arch" : "HVM64" },
    "r3.8xlarge" : { "Arch" : "HVM64" },
    "i2.xlarge" : { "Arch" : "HVM64" },
    "i2.2xlarge" : { "Arch" : "HVM64" },
    "i2.4xlarge" : { "Arch" : "HVM64" },
    "i2.8xlarge" : { "Arch" : "HVM64" },
    "d2.xlarge" : { "Arch" : "HVM64" },
    "d2.2xlarge" : { "Arch" : "HVM64" },
    "d2.4xlarge" : { "Arch" : "HVM64" },
    "d2.8xlarge" : { "Arch" : "HVM64" },
    "hi1.4xlarge" : { "Arch" : "HVM64" },
    "hs1.8xlarge" : { "Arch" : "HVM64" },
    "cr1.8xlarge" : { "Arch" : "HVM64" },
    "cc2.8xlarge" : { "Arch" : "HVM64" }
  },

  "AWSRegionArch2AMI" : {
    "us-east-1" : { "HVM64" : "ami-0ff8a91507f77f867", "HVMG2" :
"ami-0a584ac55a7631c0c"},
    "us-west-2" : { "HVM64" : "ami-a0cfeed8", "HVMG2" :
"ami-0e09505bc235aa82d"},
    "us-west-1" : { "HVM64" : "ami-0bdb828fd58c52235", "HVMG2" :
"ami-066ee5fd4a9ef77f1"},
    "eu-west-1" : { "HVM64" : "ami-047bb4163c506cd98", "HVMG2" :
"ami-0a7c483d527806435"},
    "eu-west-2" : { "HVM64" : "ami-f976839e", "HVMG2" : "NOT_SUPPORTED"},
    "eu-west-3" : { "HVM64" : "ami-0ebc281c20e89ba4b", "HVMG2" :
"NOT_SUPPORTED"},
    "eu-central-1" : { "HVM64" : "ami-0233214e13e500f77", "HVMG2" :
"ami-06223d46a6d0661c7"},

```

```

    "ap-northeast-1" : {"HVM64" : "ami-06cd52961ce9f0d85", "HVMG2" :
"ami-053cdd503598e4a9d"},
    "ap-northeast-2" : {"HVM64" : "ami-0a10b2721688ce9d2", "HVMG2" :
"NOT_SUPPORTED"},
    "ap-northeast-3" : {"HVM64" : "ami-0d98120a9fb693f07", "HVMG2" :
"NOT_SUPPORTED"},
    "ap-southeast-1" : {"HVM64" : "ami-08569b978cc4dfa10", "HVMG2" :
"ami-0be9df32ae9f92309"},
    "ap-southeast-2" : {"HVM64" : "ami-09b42976632b27e9b", "HVMG2" :
"ami-0a9ce9fecc3d1daf8"},
    "ap-south-1" : {"HVM64" : "ami-0912f71e06545ad88", "HVMG2" :
"ami-097b15e89dbdcfcf4"},
    "us-east-2" : {"HVM64" : "ami-0b59bfac6be064b78", "HVMG2" :
"NOT_SUPPORTED"},
    "ca-central-1" : {"HVM64" : "ami-0b18956f", "HVMG2" : "NOT_SUPPORTED"},
    "sa-east-1" : {"HVM64" : "ami-07b14488da8ea02a0", "HVMG2" :
"NOT_SUPPORTED"},
    "cn-north-1" : {"HVM64" : "ami-0a4eaf6c4454eda75", "HVMG2" :
"NOT_SUPPORTED"},
    "cn-northwest-1" : {"HVM64" : "ami-6b6a7d09", "HVMG2" : "NOT_SUPPORTED"}
  }
},

"Resources" : {

  "WebServerInstance": {
    "Type" : "AWS::EC2::Instance",
    "Metadata" : {
      "Comment" : "Install a simple PHP application",
      "AWS::CloudFormation::Init" : {
        "config" : {
          "packages" : {
            "yum" : {
              "httpd" : [],
              "php" : []
            }
          }
        },

        "files" : {

          "/var/www/html/index.php" : {
            "content" : { "Fn::Join" : ["", [
              "<?php\n",
              "echo '<h1>AWS CloudFormation sample PHP application</h1>';\n",

```

```

        "?>\n"
    ]]],
    "mode"      : "000644",
    "owner"     : "apache",
    "group"     : "apache"
},

"/etc/cfn/cfn-hup.conf" : {
    "content" : { "Fn::Join" : ["", [
        "[main]\n",
        "stack=", { "Ref" : "AWS::StackId" }, "\n",
        "region=", { "Ref" : "AWS::Region" }, "\n"
    ]]],
    "mode"      : "000400",
    "owner"     : "root",
    "group"     : "root"
},

"/etc/cfn/hooks.d/cfn-auto-reloader.conf" : {
    "content": { "Fn::Join" : ["", [
        "[cfn-auto-reloader-hook]\n",
        "triggers=post.update\n",
        "path=Resources.WebServerInstance.Metadata.AWS::CloudFormation::Init
\n",
        "action=/opt/aws/bin/cfn-init -s ", { "Ref" : "AWS::StackId" }, " -r
WebServerInstance ",
        " --region      ", { "Ref" :
"AWS::Region" }, "\n",
        "runas=root\n"
    ]]]
    }
},

"services" : {
    "sysvinit" : {
        "httpd"      : { "enabled" : "true", "ensureRunning" : "true" },
        "cfn-hup"    : { "enabled" : "true", "ensureRunning" : "true",
            "files" : ["/etc/cfn/cfn-hup.conf", "/etc/cfn/hooks.d/cfn-auto-
reloader.conf"]}
    }
}
}
}
}

```

```

    },

    "Properties": {
        "ImageId" : { "Fn::FindInMap" : [ "AWSRegionArch2AMI", { "Ref" :
"AWS::Region" },
                                { "Fn::FindInMap" : [ "AWSInstanceType2Arch", { "Ref" :
"InstanceType" }, "Arch" ] } ] },
        "InstanceType" : { "Ref" : "InstanceType" },
        "SecurityGroups" : [ { "Ref" : "WebServerSecurityGroup" } ],
        "UserData" : { "Fn::Base64" : { "Fn::Join" : [ "", [
            "#!/bin/bash -xe\n",
            "yum install -y aws-cfn-bootstrap\n",

            "# Install the files and packages from the metadata\n",
            "/opt/aws/bin/cfn-init -v ",
            "    --stack ", { "Ref" : "AWS::StackName" },
            "    --resource WebServerInstance ",
            "    --region ", { "Ref" : "AWS::Region" }, "\n",

            "# Start up the cfn-hup daemon to listen for changes to the Web Server
metadata\n",
            "/opt/aws/bin/cfn-hup || error_exit 'Failed to start cfn-hup'\n",

            "# Signal the status from cfn-init\n",
            "/opt/aws/bin/cfn-signal -e $? ",
            "    --stack ", { "Ref" : "AWS::StackName" },
            "    --resource WebServerInstance ",
            "    --region ", { "Ref" : "AWS::Region" }, "\n"
        ] ] ] }
    ] }
},

"CreationPolicy" : {
    "ResourceSignal" : {
        "Timeout" : "PT5M"
    }
}
},

"WebServerSecurityGroup" : {
    "Type" : "AWS::EC2::SecurityGroup",
    "Properties" : {
        "GroupDescription" : "Enable HTTP access via port 80",
        "SecurityGroupIngress" : [
            { "IpProtocol" : "tcp", "FromPort" : "80", "ToPort" : "80", "CidrIp" :
"0.0.0.0/0" }

```

```
    ]
  }
}
},

"Outputs" : {
  "WebsiteURL" : {
    "Description" : "Application URL",
    "Value" : { "Fn::Join" : [ "", [ "http://", { "Fn::GetAtt" : [ "WebServerInstance",
"PublicDnsName" ] } ] ] }
  }
}
}
```

Contoh ini menggunakan satu EC2 instans Amazon, tetapi Anda dapat menggunakan mekanisme yang sama pada solusi yang lebih kompleks yang menggunakan Elastic Load Balancing dan Amazon EC2 Auto Scaling grup untuk mengelola kumpulan server aplikasi. Namun, ada beberapa pertimbangan khusus untuk grup Auto Scaling. Untuk informasi selengkapnya, lihat [Memperbarui grup Auto Scaling](#).

Membuat tumpukan awal

Untuk keperluan contoh ini, kita akan menggunakan AWS Management Console untuk membuat tumpukan awal dari template sampel.

Warning

Menyelesaikan prosedur ini akan menggunakan AWS layanan langsung. Anda akan dikenakan tarif penggunaan standar selama layanan ini berjalan.

Untuk membuat tumpukan dari AWS Management Console

1. Salin templat sebelumnya dan simpan secara lokal pada sistem Anda sebagai file teks. Catat lokasinya karena Anda harus menggunakan file di langkah berikutnya.
2. Masuk ke CloudFormation konsol di <https://console.aws.amazon.com/cloudformation>.
3. Pilih Buat Tumpukan Baru.
4. Di panduan Membuat Tumpukan Baru, pada layar Pilih Templat, ketikkan **UpdateTutorial** di bidang Nama. Pada halaman yang sama, pilih Unggah templat ke Amazon S3 Lalu jelajahi file yang Anda unduh di langkah pertama, lalu pilih Selanjutnya.

5. Di layar Tentukan Parameter, di kotak Tipe instans, ketikkan **t1.micro**. Lalu pilih Selanjutnya.
6. Di layar Opsi, pilih Selanjutnya.
7. Di layar Tinjau, pastikan semua pengaturan sesuai keinginan Anda, lalu pilih Buat.

Setelah status tumpukan Anda CREATE_COMPLETE, tab output akan menampilkan URL dari situs web Anda. Jika Anda memilih nilai output WebsiteURL, Anda akan melihat aplikasi PHP baru Anda bekerja.

Memperbarui aplikasi

Karena kita sudah men-deploy tumpukan, mari kita perbarui aplikasi ini. Kami akan membuat perubahan sederhana pada teks yang dicetak oleh aplikasi. Untuk melakukannya, kita akan menambahkan perintah echo ke file index.php seperti yang ditunjukkan dalam snippet templat ini:

```
"WebServerInstance": {
  "Type" : "AWS::EC2::Instance",
  "Metadata" : {
    "AWS::CloudFormation::Init" : {
      "config" : {
        :

      "files" : {

        "/var/www/html/index.php" : {
          "content" : { "Fn::Join" : ["", [
            "<?php\n",
            "echo '<h1>AWS CloudFormation sample PHP application</h1>';\n",
            "echo '<p>Updated version via UpdateStack</p>';\n ",
            "?>"
          ]}},
          "mode" : "000644",
          "owner" : "apache",
          "group" : "apache"
        },

        :

      ]
    }
  },
}
```

Gunakan editor teks untuk secara manual mengedit file templat yang Anda simpan secara lokal.

Sekarang, kita akan memperbarui tumpukan.

Untuk memperbarui tumpukan dari AWS Management Console

1. Masuk ke CloudFormation konsol, di: <https://console.aws.amazon.com/cloudformation>.
2. Di CloudFormation dasbor, pilih tumpukan yang Anda buat sebelumnya, lalu pilih Update Stack.
3. Di panduan Memperbarui Tumpukan, di layar Pilih templat, pilih Unggah templat ke Amazon S3, pilih templat yang dimodifikasi, kemudian pilih Selanjutnya.
4. Di layar Opsi, pilih Selanjutnya.
5. Pilih Selanjutnya karena tumpukan tidak memiliki kebijakan tumpukan. Semua sumber daya dapat diperbarui tanpa kebijakan timpa.
6. Di layar Tinjau, pastikan semua pengaturan sesuai keinginan Anda, lalu pilih Perbarui.

Jika Anda memperbarui tumpukan dari AWS Management Console, Anda akan melihat bahwa parameter yang digunakan untuk membuat tumpukan awal diisi sebelumnya pada halaman Parameter wizard Update Stack. Jika Anda menggunakan perintah `update-stack`, pastikan untuk mengetikkan nilai yang sama untuk parameter yang Anda gunakan awalnya untuk membuat tumpukan.

Ketika tumpukan Anda dalam status `UPDATE_COMPLETE`, Anda dapat memilih nilai output `WebsiteURL` lagi untuk memastikan bahwa perubahan di aplikasi Anda telah berlaku. Secara default, daemon `cfn-hup` berjalan setiap 15 menit, sehingga mungkin diperlukan waktu hingga 15 menit untuk hingga perubahan diterapkan di aplikasi setelah tumpukan telah diperbarui.

Untuk melihat kumpulan sumber daya yang diperbarui, buka CloudFormation konsol. Di tab Peristiwa, lihat peristiwa tumpukan. Dalam kasus khusus ini, metadata untuk EC2 instans Amazon `WebServerInstance` diperbarui, yang CloudFormation menyebabkan juga mengevaluasi kembali resource (`WebServerSecurityGroup`) lainnya untuk memastikan bahwa tidak ada perubahan lain. Tak satu pun dari sumber daya tumpukan lainnya yang dimodifikasi. CloudFormation hanya akan memperbarui sumber daya di tumpukan yang dipengaruhi oleh perubahan apa pun pada tumpukan. Perubahan tersebut dapat bersifat langsung, seperti perubahan properti atau metadata, atau dapat disebabkan oleh dependensi atau aliran data melalui `Ref`, `GetAtt`, atau fungsi template intrinsik lainnya. Untuk informasi lebih lanjut, lihat Referensi [fungsi intrinsik](#).

Pembaruan sederhana ini menggambarkan prosesnya; Namun, Anda dapat membuat perubahan yang jauh lebih kompleks pada file dan paket yang digunakan ke instance Amazon EC2 Anda. Misalnya, Anda mungkin memutuskan bahwa Anda perlu menambahkan MySQL ke instans, bersama

dengan dukungan PHP untuk MySQL. Untuk melakukannya, cukup tambahkan paket dan file tambahan bersama dengan layanan tambahan apa pun ke konfigurasi, lalu perbarui tumpukan untuk menerapkan perubahan. Dalam snippet templat berikut, perubahan disorot dalam warna merah:

```
"WebServerInstance": {
  "Type" : "AWS::EC2::Instance",
  "Metadata" : {
    "Comment" : "Install a simple PHP application",
    "AWS::CloudFormation::Init" : {
      "config" : {
        "packages" : {
          "yum" : {
            "httpd"           : [],
            "php"             : [],
            "php-mysql"       : [],
            "mysql-server"    : [],
            "mysql-libs"     : [],
            "mysql"           : []
          }
        },
        :

        "services" : {
          "sysvinit" : {
            "httpd"   : { "enabled" : "true", "ensureRunning" : "true" },
            "cfn-hup" : { "enabled" : "true", "ensureRunning" : "true",
              "files" : ["/etc/cfn/cfn-hup.conf", "/etc/cfn/hooks.d/cfn-auto-
reloader.conf"]},
            "mysqld"   : { "enabled" : "true", "ensureRunning" : "true" }
          }
        }
      }
    }
  },
  "Properties": {
    :
  }
}
```

Anda dapat memperbarui CloudFormation metadata untuk memperbarui ke versi baru dari paket yang digunakan oleh aplikasi. Di contoh sebelumnya, properti versi untuk setiap paket kosong, menunjukkan bahwa cfn-init harus menginstal versi terbaru dari paket.

```
"packages" : {
  "yum" : {
    "httpd"      : [],
    "php"        : []
  }
}
```

Anda dapat secara opsional menentukan string versi untuk paket. Jika Anda mengubah string versi dalam panggilan tumpukan pembaruan berikutnya, versi baru dari paket akan di-deploy. Berikut adalah contoh penggunaan nomor versi untuk RubyGems paket. Setiap paket yang mendukung versioning dapat memiliki versi tertentu.

```
"packages" : {
  "rubygems" : {
    "mysql"      : [],
    "rubygems-update" : ["1.6.2"],
    "rake"       : ["0.8.7"],
    "rails"      : ["2.3.11"]
  }
}
```

Memperbarui grup Auto Scaling

Jika Anda menggunakan grup Auto Scaling di template Anda, sebagai lawan dari sumber daya EC2 instans Amazon, memperbarui aplikasi akan bekerja dengan cara yang persis sama; namun, CloudFormation tidak menyediakan sinkronisasi atau serialisasi apa pun di seluruh EC2 instans Amazon dalam grup Auto Scaling. Daemon cfn-hup pada setiap host akan berjalan sendiri dan memperbarui aplikasi sesuai jadwalnya sendiri. Ketika Anda menggunakan cfn-hup untuk memperbarui konfigurasi pada instans, setiap instans akan menjalankan kait cfn-hup sesuai jadwalnya sendiri; tidak ada koordinasi di antara instans di tumpukan. Anda harus mempertimbangkan hal berikut:

- Jika perubahan cfn-hup berjalan di semua EC2 instans Amazon di grup Auto Scaling secara bersamaan, layanan Anda mungkin tidak tersedia selama pembaruan.
- Jika perubahan cfn-hup berjalan pada waktu yang berbeda, versi lama dan baru dari perangkat lunak mungkin berjalan pada waktu yang sama.

Untuk menghindari masalah ini, pertimbangkan untuk memaksa pembaruan bergulir pada instans Anda di grup Auto Scaling. Untuk informasi lebih lanjut, lihat [Atribut UpdatePolicy](#).

Mengubah properti sumber daya

Dengan CloudFormation, Anda dapat mengubah properti sumber daya yang ada di tumpukan. Bagian berikut menjelaskan berbagai pembaruan yang memecahkan masalah tertentu; tetapi, properti apa pun dari sumber daya apa pun yang mendukung pembaruan dalam tumpukan dapat dimodifikasi seperlunya.

Memperbarui tipe instans

Tumpukan yang kami buat sejauh ini menggunakan instance Amazon EC2 t1.micro. Misalkan situs web Anda yang baru dibuat mendapatkan lebih banyak lalu lintas daripada yang dapat ditangani oleh instans t1.micro, dan sekarang Anda ingin pindah ke jenis instans Amazon EC2 m1.small. Jika arsitektur tipe instans berubah, instans akan dibuat dengan AMI yang berbeda. Jika Anda memeriksa pemetaan di template, Anda akan melihat bahwa t1.micro dan m1.small adalah arsitektur yang sama dan menggunakan Amazon Linux yang sama. AMIs

```
"Mappings" : {
  "AWSInstanceType2Arch" : {
    "t1.micro"      : { "Arch" : "HVM64" },
    "t2.nano"      : { "Arch" : "HVM64" },
    "t2.micro"      : { "Arch" : "HVM64" },
    "t2.small"     : { "Arch" : "HVM64" },
    "t2.medium"    : { "Arch" : "HVM64" },
    "t2.large"     : { "Arch" : "HVM64" },
    "m1.small"     : { "Arch" : "HVM64" },
    "m1.medium"    : { "Arch" : "HVM64" },
    "m1.large"     : { "Arch" : "HVM64" },
    "m1.xlarge"    : { "Arch" : "HVM64" },
    "m2.xlarge"    : { "Arch" : "HVM64" },
    "m2.2xlarge"   : { "Arch" : "HVM64" },
    "m2.4xlarge"   : { "Arch" : "HVM64" },
    "m3.medium"    : { "Arch" : "HVM64" },
    "m3.large"     : { "Arch" : "HVM64" },
    "m3.xlarge"    : { "Arch" : "HVM64" },
    "m3.2xlarge"   : { "Arch" : "HVM64" },
    "m4.large"     : { "Arch" : "HVM64" },
    "m4.xlarge"    : { "Arch" : "HVM64" },
    "m4.2xlarge"   : { "Arch" : "HVM64" },
    "m4.4xlarge"   : { "Arch" : "HVM64" },
```

```

    "m4.10xlarge" : { "Arch" : "HVM64" },
    "c1.medium"   : { "Arch" : "HVM64" },
    "c1.xlarge"   : { "Arch" : "HVM64" },
    "c3.large"     : { "Arch" : "HVM64" },
    "c3.xlarge"    : { "Arch" : "HVM64" },
    "c3.2xlarge"   : { "Arch" : "HVM64" },
    "c3.4xlarge"   : { "Arch" : "HVM64" },
    "c3.8xlarge"   : { "Arch" : "HVM64" },
    "c4.large"     : { "Arch" : "HVM64" },
    "c4.xlarge"    : { "Arch" : "HVM64" },
    "c4.2xlarge"   : { "Arch" : "HVM64" },
    "c4.4xlarge"   : { "Arch" : "HVM64" },
    "c4.8xlarge"   : { "Arch" : "HVM64" },
    "g2.2xlarge"   : { "Arch" : "HVMG2" },
    "g2.8xlarge"   : { "Arch" : "HVMG2" },
    "r3.large"     : { "Arch" : "HVM64" },
    "r3.xlarge"    : { "Arch" : "HVM64" },
    "r3.2xlarge"   : { "Arch" : "HVM64" },
    "r3.4xlarge"   : { "Arch" : "HVM64" },
    "r3.8xlarge"   : { "Arch" : "HVM64" },
    "i2.xlarge"    : { "Arch" : "HVM64" },
    "i2.2xlarge"   : { "Arch" : "HVM64" },
    "i2.4xlarge"   : { "Arch" : "HVM64" },
    "i2.8xlarge"   : { "Arch" : "HVM64" },
    "d2.xlarge"    : { "Arch" : "HVM64" },
    "d2.2xlarge"   : { "Arch" : "HVM64" },
    "d2.4xlarge"   : { "Arch" : "HVM64" },
    "d2.8xlarge"   : { "Arch" : "HVM64" },
    "hi1.4xlarge"  : { "Arch" : "HVM64" },
    "hs1.8xlarge"  : { "Arch" : "HVM64" },
    "cr1.8xlarge"  : { "Arch" : "HVM64" },
    "cc2.8xlarge"  : { "Arch" : "HVM64" }
  },
  "AWSRegionArch2AMI" : {
    "us-east-1"       : {"HVM64" : "ami-0ff8a91507f77f867", "HVMG2" :
"ami-0a584ac55a7631c0c"},
    "us-west-2"       : {"HVM64" : "ami-a0cfeed8", "HVMG2" :
"ami-0e09505bc235aa82d"},
    "us-west-1"       : {"HVM64" : "ami-0bdb828fd58c52235", "HVMG2" :
"ami-066ee5fd4a9ef77f1"},
    "eu-west-1"       : {"HVM64" : "ami-047bb4163c506cd98", "HVMG2" :
"ami-0a7c483d527806435"},
    "eu-west-2"       : {"HVM64" : "ami-f976839e", "HVMG2" : "NOT_SUPPORTED"},

```

```

    "eu-west-3"      : {"HVM64" : "ami-0ebc281c20e89ba4b", "HVMG2" :
"NOT_SUPPORTED"},
    "eu-central-1"   : {"HVM64" : "ami-0233214e13e500f77", "HVMG2" :
"ami-06223d46a6d0661c7"},
    "ap-northeast-1" : {"HVM64" : "ami-06cd52961ce9f0d85", "HVMG2" :
"ami-053cdd503598e4a9d"},
    "ap-northeast-2" : {"HVM64" : "ami-0a10b2721688ce9d2", "HVMG2" :
"NOT_SUPPORTED"},
    "ap-northeast-3" : {"HVM64" : "ami-0d98120a9fb693f07", "HVMG2" :
"NOT_SUPPORTED"},
    "ap-southeast-1" : {"HVM64" : "ami-08569b978cc4dfa10", "HVMG2" :
"ami-0be9df32ae9f92309"},
    "ap-southeast-2" : {"HVM64" : "ami-09b42976632b27e9b", "HVMG2" :
"ami-0a9ce9fecc3d1daf8"},
    "ap-south-1"     : {"HVM64" : "ami-0912f71e06545ad88", "HVMG2" :
"ami-097b15e89dbdcfcf4"},
    "us-east-2"      : {"HVM64" : "ami-0b59bfac6be064b78", "HVMG2" :
"NOT_SUPPORTED"},
    "ca-central-1"   : {"HVM64" : "ami-0b18956f", "HVMG2" : "NOT_SUPPORTED"},
    "sa-east-1"      : {"HVM64" : "ami-07b14488da8ea02a0", "HVMG2" :
"NOT_SUPPORTED"},
    "cn-north-1"     : {"HVM64" : "ami-0a4eaf6c4454eda75", "HVMG2" :
"NOT_SUPPORTED"},
    "cn-northwest-1" : {"HVM64" : "ami-6b6a7d09", "HVMG2" : "NOT_SUPPORTED"}
  }

```

Mari kita gunakan templat yang kita modifikasi di bagian sebelumnya untuk mengubah tipe instans. Karena InstanceType merupakan parameter input ke template, kita tidak perlu memodifikasi template; kita dapat mengubah nilai parameter di wizard Stack Update, pada halaman Specify Parameters.

Untuk memperbarui tumpukan dari AWS Management Console

1. Masuk ke CloudFormation konsol di <https://console.aws.amazon.com/cloudformation>.
2. Di CloudFormation dasbor, pilih tumpukan yang Anda buat sebelumnya, lalu pilih Update Stack.
3. Di panduan Memperbarui Tumpukan, di templat Pilih Templat pilih Gunakan templat saat ini, lalu pilih Selanjutnya.

Halaman Tentukan Rincian muncul dengan parameter yang digunakan untuk membuat tumpukan awal yang telah diisi sebelumnya di bagian Tentukan Parameter.

4. Ubah nilai kotak InstanceTypeteks dari `t1.micro` ke `m1.small`. Lalu, pilih Selanjutnya.
5. Di layar Opsi, pilih Selanjutnya.

6. Pilih Selanjutnya karena tumpukan tidak memiliki kebijakan tumpukan. Semua sumber daya dapat diperbarui tanpa kebijakan tumpukan.
7. Di layar Tinjau, pastikan semua pengaturan sesuai keinginan Anda, lalu pilih Perbarui.

Anda dapat secara dinamis mengubah jenis instans dari instans EC2 Amazon yang didukung EBS dengan memulai dan menghentikan instans. CloudFormation mencoba mengoptimalkan perubahan dengan memperbarui jenis instance dan memulai ulang instance, sehingga ID instance tidak berubah. Ketika instans dimulai ulang, tetapi, alamat IP publik dari instans tidak berubah. Untuk memastikan bahwa alamat IP Elastis terikat dengan benar setelah perubahan, juga CloudFormation akan memperbarui alamat IP Elastis. Anda dapat melihat perubahan di CloudFormation konsol pada tab Acara.

Untuk memeriksa jenis instance dari AWS Management Console, buka EC2 konsol Amazon, dan temukan instance Anda di sana.

Perbarui AMI pada EC2 instans Amazon

Sekarang mari kita lihat bagaimana kita dapat mengubah Amazon Machine Image (AMI) yang berjalan pada instans. Kami akan memulai perubahan AMI dengan memperbarui tumpukan untuk menggunakan jenis EC2 instans Amazon baru, seperti t2.medium, yang merupakan tipe instance. HVM64

Seperti di bagian sebelumnya, kita akan menggunakan templat yang ada untuk mengubah tipe instans yang digunakan oleh contoh tumpukan kita. Di panduan Memperbarui Tumpukan, di halaman Tentukan Parameter, ubah nilai dari Tipe Instans.

Dalam hal ini, kita tidak bisa hanya memulai dan menghentikan instans untuk memodifikasi AMI; CloudFormation menganggap ini perubahan ke properti sumber daya yang tidak dapat diubah. Untuk membuat perubahan pada properti yang tidak dapat diubah, CloudFormation harus meluncurkan sumber daya pengganti, dalam hal ini EC2 instance Amazon baru yang menjalankan AMI baru.

Setelah instance baru berjalan, CloudFormation perbarui sumber daya lain di tumpukan untuk menunjuk ke sumber daya baru. Ketika semua sumber daya baru dibuat, sumber daya lama dihapus, sebuah proses yang dikenal sebagai UPDATE_CLEANUP. Kali ini, Anda akan melihat bahwa ID instans dan URL aplikasi dari instans di dalam tumpukan telah berubah sebagai hasil dari pembaruan. Peristiwa dalam tabel Peristiwa berisi deskripsi "Pembaruan yang diminta memiliki perubahan pada properti yang tidak dapat diubah dan karenanya membuat sumber daya fisik baru" untuk menunjukkan bahwa sumber daya telah diganti.

Jika Anda memiliki kode aplikasi yang ditulis ke dalam AMI yang ingin Anda perbarui, Anda dapat menggunakan mekanisme pembaruan tumpukan yang sama untuk memperbarui AMI untuk memuat aplikasi baru Anda.

Untuk memperbarui AMI untuk instans di tumpukan Anda

1. Buat AMIs yang baru berisi perubahan aplikasi atau sistem operasi Anda. Untuk informasi selengkapnya, buka [Buat AMI yang didukung Amazon EBS-backed di Panduan Pengguna Amazon. EC2](#)
2. Perbarui template Anda untuk menggabungkan AMI baru IDs.
3. Perbarui tumpukan, baik dari AWS Management Console seperti yang dijelaskan dalam [Memperbarui aplikasi](#) atau dengan menggunakan AWS perintah [update-stack](#).

Saat Anda memperbarui tumpukan, CloudFormation mendeteksi bahwa ID AMI telah berubah, dan kemudian memicu pembaruan tumpukan dengan cara yang sama seperti kami memulai yang di atas.

Perbarui konfigurasi EC2 peluncuran Amazon untuk grup Auto Scaling

Jika Anda menggunakan grup Auto Scaling daripada EC2 instans Amazon, proses memperbarui instans yang sedang berjalan sedikit berbeda. Dengan sumber daya Auto Scaling, konfigurasi instans Amazon, seperti jenis EC2 instans atau ID AMI dienkapsulasi dalam konfigurasi peluncuran Auto Scaling. Anda dapat membuat perubahan pada konfigurasi peluncuran dengan cara yang sama seperti kami membuat perubahan pada sumber daya EC2 instans Amazon di bagian sebelumnya. Namun, mengubah konfigurasi peluncuran tidak memengaruhi EC2 instans Amazon yang sedang berjalan di grup Auto Scaling. Konfigurasi peluncuran yang diperbarui hanya berlaku untuk instance baru yang dibuat setelah pembaruan.

Jika Anda ingin menerapkan perubahan ke konfigurasi peluncuran di semua instans dalam grup Auto Scaling, Anda dapat menggunakan atribut update. Untuk informasi lebih lanjut, lihat [Atribut UpdatePolicy](#).

Menambahkan properti sumber daya

Sejauh ini, kita telah melihat perubahan properti sumber daya yang ada dalam templat. Anda juga dapat menambahkan properti yang awalnya tidak ditentukan dalam templat. Untuk mengilustrasikannya, kami akan menambahkan EC2 key pair Amazon ke EC2 instance yang ada dan kemudian membuka port 22 di Amazon EC2 Security Group sehingga Anda dapat menggunakan Secure Shell (SSH) untuk mengakses instans.

Menambahkan pasangan kunci ke sebuah instans

Untuk menambahkan akses SSH ke instans Amazon EC2 yang ada

1. Tambahkan dua parameter tambahan ke template untuk meneruskan nama Amazon EC2 key pair dan lokasi SSH yang ada.

```
"Parameters" : {

  "KeyName" : {
    "Description" : "Name of an existing Amazon EC2 key pair for SSH access",
    "Type": "AWS::EC2::KeyPair::KeyName"
  },
  "SSHLocation" : {
    "Description" : " The IP address range that can be used to SSH to the EC2
instances",
    "Type": "String",
    "MinLength": "9",
    "MaxLength": "18",
    "Default": "0.0.0.0/0",
    "AllowedPattern": "(\\d{1,3})\\. (\\d{1,3})\\. (\\d{1,3})\\. (\\d{1,3})/ (\\d{1,2})",
    "ConstraintDescription": "must be a valid IP CIDR range of the form x.x.x.x/
x."
  }
}
```

2. Tambahkan KeyName properti ke EC2 instance Amazon.

```
"WebServerInstance": {
  "Type" : "AWS::EC2::Instance",
  :
  "Properties": {
    :
    "KeyName" : { "Ref" : "KeyName" },
    :
  }
},
```

3. Tambahkan port 22 dan lokasi SSH ke aturan masuk untuk grup EC2 keamanan Amazon.

```
"WebServerSecurityGroup" : {
```

```

    "Type" : "AWS::EC2::SecurityGroup",
    "Properties" : {
      "GroupDescription" : "Enable HTTP and SSH",
      "SecurityGroupIngress" : [
        { "IpProtocol" : "tcp", "FromPort" : "22", "ToPort" : "22", "CidrIp" :
{ "Ref" : "SSHLocation" } },
        { "IpProtocol" : "tcp", "FromPort" : "80", "ToPort" : "80", "CidrIp" :
"0.0.0.0/0" }
      ]
    }
  },

```

4. Perbarui tumpukan, baik dari AWS Management Console seperti yang dijelaskan dalam [Memperbarui aplikasi](#) atau dengan menggunakan AWS perintah [update-stack](#).

Mengubah sumber daya tumpukan

Kebutuhan aplikasi dapat berubah seiring waktu, CloudFormation memungkinkan Anda mengubah kumpulan sumber daya yang membentuk tumpukan. Untuk mendemonstrasikannya, kita akan mengambil aplikasi instans tunggal dari [Menambahkan properti sumber daya](#) dan mengonversinya menjadi aplikasi yang diskalakan secara otomatis dan memiliki beban seimbang dengan memperbarui tumpukan.

Ini akan membuat satu aplikasi PHP instans menggunakan alamat IP Elastis. Kita sekarang akan mengubah aplikasi menjadi aplikasi yang sangat tersedia, diskalakan otomatis, memiliki beban seimbang dengan mengubah sumber dayanya selama pembaruan.

1. Tambahkan sumber daya Elastic Load Balancer.

```

"ElasticLoadBalancer" : {
  "Type" : "AWS::ElasticLoadBalancing::LoadBalancer",
  "Properties" : {
    "CrossZone" : "true",
    "AvailabilityZones" : { "Fn::GetAZs" : "" },
    "LBCookieStickinessPolicy" : [ {
      "PolicyName" : "CookieBasedPolicy",
      "CookieExpirationPeriod" : "30"
    } ],
    "Listeners" : [ {
      "LoadBalancerPort" : "80",
      "InstancePort" : "80",
      "Protocol" : "HTTP",

```

```

    "PolicyNames" : [ "CookieBasedPolicy" ]
  } ],
  "HealthCheck" : {
    "Target" : "HTTP:80/",
    "HealthyThreshold" : "2",
    "UnhealthyThreshold" : "5",
    "Interval" : "10",
    "Timeout" : "5"
  }
}
}

```

2. Mengkonversi EC2 instance dalam template menjadi Auto Scaling Launch Configuration. Propertinya serupa, jadi kita hanya perlu mengubah nama tipe dari:

```

"WebServerInstance": {
  "Type" : "AWS::EC2::Instance",

```

ke:

```

"LaunchConfig": {
  "Type" : "AWS::AutoScaling::LaunchConfiguration",

```

Untuk kejelasan dalam template, kami mengubah nama sumber daya dari `WebServerInstance` menjadi `LaunchConfig`, jadi Anda harus memperbarui nama sumber daya yang direferensikan oleh `cfn-init` dan `cfn-hup` (cukup cari `WebServerInstance` dan ganti dengan `LaunchConfig`, kecuali untuk `cfn-signal`). Untuk `cfn-signal`, Anda harus memberi sinyal pada grup Auto Scaling `WebServerGroup` () bukan instance, seperti yang ditunjukkan pada cuplikan berikut:

```

"# Signal the status from cfn-init\n",
"/opt/aws/bin/cfn-signal -e $? ",
"      --stack ", { "Ref" : "AWS::StackName" },
"      --resource WebServerGroup ",
"      --region ", { "Ref" : "AWS::Region" }, "\n"

```

3. Tambahkan sumber daya grup Auto Scaling.

```

"WebServerGroup" : {
  "Type" : "AWS::AutoScaling::AutoScalingGroup",

```

```

"Properties" : {
  "AvailabilityZones" : { "Fn::GetAZs" : "" },
  "LaunchConfigurationName" : { "Ref" : "LaunchConfig" },
  "MinSize" : "1",
  "DesiredCapacity" : "1",
  "MaxSize" : "5",
  "LoadBalancerNames" : [ { "Ref" : "ElasticLoadBalancer" } ]
},
"CreationPolicy" : {
  "ResourceSignal" : {
    "Timeout" : "PT15M"
  }
},
"UpdatePolicy": {
  "AutoScalingRollingUpdate": {
    "MinInstancesInService": "1",
    "MaxBatchSize": "1",
    "PauseTime" : "PT15M",
    "WaitOnResourceSignals": "true"
  }
}
}

```

4. Perbarui definisi Grup Keamanan untuk mengunci lalu lintas untuk instans dari penyeimbang beban.

```

"WebServerSecurityGroup" : {
  "Type" : "AWS::EC2::SecurityGroup",
  "Properties" : {
    "GroupDescription" : "Enable HTTP access via port 80 locked down to the ELB and SSH access",
    "SecurityGroupIngress" : [
      { "IpProtocol" : "tcp", "FromPort" : "80", "ToPort" : "80",
        "SourceSecurityGroupOwnerId" : { "Fn::GetAtt" : ["ElasticLoadBalancer", "SourceSecurityGroup.OwnerAlias"] },
        "SourceSecurityGroupName" : { "Fn::GetAtt" : ["ElasticLoadBalancer", "SourceSecurityGroup.GroupName"] },
        { "IpProtocol" : "tcp", "FromPort" : "22", "ToPort" : "22", "CidrIp" : { "Ref" : "SSHLocation" } }
      ]
    }
  }
}

```

5. Perbarui Output untuk mengembalikan Nama DNS Elastic Load Balancer sebagai lokasi aplikasi dari:

```
"WebsiteURL" : {
  "Value" : { "Fn::Join" : [ "", [ "http://",
    { "Fn::GetAtt" : [ "WebServerInstance", "PublicDnsName" ] } ] ] },
  "Description" : "Application URL"
}
```

ke:

```
"WebsiteURL" : {
  "Value" : { "Fn::Join" : [ "", [ "http://",
    { "Fn::GetAtt" : [ "ElasticLoadBalancer", "DNSName" ] } ] ] },
  "Description" : "Application URL"
}
```

Untuk referensi, contoh berikut menunjukkan templat lengkap. Jika Anda menggunakan template ini untuk memperbarui tumpukan, Anda akan mengonversi aplikasi instance tunggal Anda yang sederhana menjadi aplikasi multi-AZ, berskala otomatis, dan load balanced yang sangat tersedia. Hanya sumber daya yang perlu diperbarui yang akan diubah, jadi misalnya ada penyimpanan data untuk aplikasi ini, data akan tetap utuh. Sekarang, Anda dapat menggunakannya CloudFormation untuk menumbuhkan atau meningkatkan tumpukan Anda saat kebutuhan Anda berubah.

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

  "Description" : "AWS CloudFormation Sample Template: Sample template that can be used
to test EC2 updates. **WARNING** This template creates an Amazon Ec2 Instance. You
will be billed for the AWS resources used if you create a stack from this template.",

  "Parameters" : {

    "KeyName": {
      "Description" : "Name of an existing EC2 KeyPair to enable SSH access to the
instance",
      "Type": "AWS::EC2::KeyPair::KeyName",
      "ConstraintDescription" : "must be the name of an existing EC2 KeyPair."
    }
  }
}
```

```

    },

    "SSHLocation" : {
        "Description" : " The IP address range that can be used to SSH to the EC2
instances",
        "Type": "String",
        "MinLength": "9",
        "MaxLength": "18",
        "Default": "0.0.0.0/0",
        "AllowedPattern": "(\\d{1,3})\\. (\\d{1,3})\\. (\\d{1,3})\\. (\\d{1,3})/(\\d{1,2})",
        "ConstraintDescription": "must be a valid IP CIDR range of the form x.x.x.x/x."
    },

    "InstanceType" : {
        "Description" : "WebServer EC2 instance type",
        "Type" : "String",
        "Default" : "t2.small",
        "AllowedValues" : [
            "t1.micro",
            "t2.nano",
            "t2.micro",
            "t2.small",
            "t2.medium",
            "t2.large",
            "m1.small",
            "m1.medium",
            "m1.large",
            "m1.xlarge",
            "m2.xlarge",
            "m2.2xlarge",
            "m2.4xlarge",
            "m3.medium",
            "m3.large",
            "m3.xlarge",
            "m3.2xlarge",
            "m4.large",
            "m4.xlarge",
            "m4.2xlarge",
            "m4.4xlarge",
            "m4.10xlarge",
            "c1.medium",
            "c1.xlarge",
            "c3.large",
            "c3.xlarge",

```

```

        "c3.2xlarge",
        "c3.4xlarge",
        "c3.8xlarge",
        "c4.large",
        "c4.xlarge",
        "c4.2xlarge",
        "c4.4xlarge",
        "c4.8xlarge",
        "g2.2xlarge",
        "g2.8xlarge",
        "r3.large",
        "r3.xlarge",
        "r3.2xlarge",
        "r3.4xlarge",
        "r3.8xlarge",
        "i2.xlarge",
        "i2.2xlarge",
        "i2.4xlarge",
        "i2.8xlarge",
        "d2.xlarge",
        "d2.2xlarge",
        "d2.4xlarge",
        "d2.8xlarge",
        "hi1.4xlarge",
        "hs1.8xlarge",
        "cr1.8xlarge",
        "cc2.8xlarge",
        "cg1.4xlarge"
    ],
    "ConstraintDescription" : "must be a valid EC2 instance type."
}
},

"Mappings" : {
    "AWSInstanceType2Arch" : {
        "t1.micro"      : { "Arch" : "HVM64" },
        "t2.nano"      : { "Arch" : "HVM64" },
        "t2.micro"      : { "Arch" : "HVM64" },
        "t2.small"     : { "Arch" : "HVM64" },
        "t2.medium"    : { "Arch" : "HVM64" },
        "t2.large"     : { "Arch" : "HVM64" },
        "m1.small"     : { "Arch" : "HVM64" },
        "m1.medium"    : { "Arch" : "HVM64" },
        "m1.large"     : { "Arch" : "HVM64" },
    },

```



```
"m1.xlarge" : { "Arch" : "HVM64" },
"m2.xlarge" : { "Arch" : "HVM64" },
"m2.2xlarge" : { "Arch" : "HVM64" },
"m2.4xlarge" : { "Arch" : "HVM64" },
"m3.medium" : { "Arch" : "HVM64" },
"m3.large" : { "Arch" : "HVM64" },
"m3.xlarge" : { "Arch" : "HVM64" },
"m3.2xlarge" : { "Arch" : "HVM64" },
"m4.large" : { "Arch" : "HVM64" },
"m4.xlarge" : { "Arch" : "HVM64" },
"m4.2xlarge" : { "Arch" : "HVM64" },
"m4.4xlarge" : { "Arch" : "HVM64" },
"m4.10xlarge" : { "Arch" : "HVM64" },
"c1.medium" : { "Arch" : "HVM64" },
"c1.xlarge" : { "Arch" : "HVM64" },
"c3.large" : { "Arch" : "HVM64" },
"c3.xlarge" : { "Arch" : "HVM64" },
"c3.2xlarge" : { "Arch" : "HVM64" },
"c3.4xlarge" : { "Arch" : "HVM64" },
"c3.8xlarge" : { "Arch" : "HVM64" },
"c4.large" : { "Arch" : "HVM64" },
"c4.xlarge" : { "Arch" : "HVM64" },
"c4.2xlarge" : { "Arch" : "HVM64" },
"c4.4xlarge" : { "Arch" : "HVM64" },
"c4.8xlarge" : { "Arch" : "HVM64" },
"g2.2xlarge" : { "Arch" : "HVMG2" },
"g2.8xlarge" : { "Arch" : "HVMG2" },
"r3.large" : { "Arch" : "HVM64" },
"r3.xlarge" : { "Arch" : "HVM64" },
"r3.2xlarge" : { "Arch" : "HVM64" },
"r3.4xlarge" : { "Arch" : "HVM64" },
"r3.8xlarge" : { "Arch" : "HVM64" },
"i2.xlarge" : { "Arch" : "HVM64" },
"i2.2xlarge" : { "Arch" : "HVM64" },
"i2.4xlarge" : { "Arch" : "HVM64" },
"i2.8xlarge" : { "Arch" : "HVM64" },
"d2.xlarge" : { "Arch" : "HVM64" },
"d2.2xlarge" : { "Arch" : "HVM64" },
"d2.4xlarge" : { "Arch" : "HVM64" },
"d2.8xlarge" : { "Arch" : "HVM64" },
"hi1.4xlarge" : { "Arch" : "HVM64" },
"hs1.8xlarge" : { "Arch" : "HVM64" },
"cr1.8xlarge" : { "Arch" : "HVM64" },
"cc2.8xlarge" : { "Arch" : "HVM64" }
```

```

    },
    "AWSRegionArch2AMI" : {
        "us-east-1"      : {"HVM64" : "ami-0ff8a91507f77f867", "HVMG2" :
"ami-0a584ac55a7631c0c"},
        "us-west-2"      : {"HVM64" : "ami-a0cfeed8", "HVMG2" :
"ami-0e09505bc235aa82d"},
        "us-west-1"      : {"HVM64" : "ami-0bdb828fd58c52235", "HVMG2" :
"ami-066ee5fd4a9ef77f1"},
        "eu-west-1"      : {"HVM64" : "ami-047bb4163c506cd98", "HVMG2" :
"ami-0a7c483d527806435"},
        "eu-west-2"      : {"HVM64" : "ami-f976839e", "HVMG2" : "NOT_SUPPORTED"},
        "eu-west-3"      : {"HVM64" : "ami-0ebc281c20e89ba4b", "HVMG2" :
"NOT_SUPPORTED"},
        "eu-central-1"   : {"HVM64" : "ami-0233214e13e500f77", "HVMG2" :
"ami-06223d46a6d0661c7"},
        "ap-northeast-1" : {"HVM64" : "ami-06cd52961ce9f0d85", "HVMG2" :
"ami-053cdd503598e4a9d"},
        "ap-northeast-2" : {"HVM64" : "ami-0a10b2721688ce9d2", "HVMG2" :
"NOT_SUPPORTED"},
        "ap-northeast-3" : {"HVM64" : "ami-0d98120a9fb693f07", "HVMG2" :
"NOT_SUPPORTED"},
        "ap-southeast-1" : {"HVM64" : "ami-08569b978cc4dfa10", "HVMG2" :
"ami-0be9df32ae9f92309"},
        "ap-southeast-2" : {"HVM64" : "ami-09b42976632b27e9b", "HVMG2" :
"ami-0a9ce9fecc3d1daf8"},
        "ap-south-1"     : {"HVM64" : "ami-0912f71e06545ad88", "HVMG2" :
"ami-097b15e89dbdcfcf4"},
        "us-east-2"      : {"HVM64" : "ami-0b59bfac6be064b78", "HVMG2" :
"NOT_SUPPORTED"},
        "ca-central-1"   : {"HVM64" : "ami-0b18956f", "HVMG2" : "NOT_SUPPORTED"},
        "sa-east-1"      : {"HVM64" : "ami-07b14488da8ea02a0", "HVMG2" :
"NOT_SUPPORTED"},
        "cn-north-1"     : {"HVM64" : "ami-0a4eaf6c4454eda75", "HVMG2" :
"NOT_SUPPORTED"},
        "cn-northwest-1" : {"HVM64" : "ami-6b6a7d09", "HVMG2" : "NOT_SUPPORTED"}
    }
},

"Resources" : {

    "ElasticLoadBalancer" : {
        "Type" : "AWS::ElasticLoadBalancing::LoadBalancer",
        "Properties" : {
            "CrossZone" : "true",

```

```

    "AvailabilityZones" : { "Fn::GetAZs" : "" },
    "LBCookieStickinessPolicy" : [ {
      "PolicyName" : "CookieBasedPolicy",
      "CookieExpirationPeriod" : "30"
    } ],
    "Listeners" : [ {
      "LoadBalancerPort" : "80",
      "InstancePort" : "80",
      "Protocol" : "HTTP",
      "PolicyNames" : [ "CookieBasedPolicy" ]
    } ],
    "HealthCheck" : {
      "Target" : "HTTP:80/",
      "HealthyThreshold" : "2",
      "UnhealthyThreshold" : "5",
      "Interval" : "10",
      "Timeout" : "5"
    }
  }
},

"WebServerGroup" : {
  "Type" : "AWS::AutoScaling::AutoScalingGroup",
  "Properties" : {
    "AvailabilityZones" : { "Fn::GetAZs" : "" },
    "LaunchConfigurationName" : { "Ref" : "LaunchConfig" },
    "MinSize" : "1",
    "DesiredCapacity" : "1",
    "MaxSize" : "5",
    "LoadBalancerNames" : [ { "Ref" : "ElasticLoadBalancer" } ]
  },
  "CreationPolicy" : {
    "ResourceSignal" : {
      "Timeout" : "PT15M"
    }
  },
  "UpdatePolicy": {
    "AutoScalingRollingUpdate": {
      "MinInstancesInService": "1",
      "MaxBatchSize": "1",
      "PauseTime" : "PT15M",
      "WaitOnResourceSignals": "true"
    }
  }
}

```

```

},

"LaunchConfig": {
  "Type" : "AWS::AutoScaling::LaunchConfiguration",
  "Metadata" : {
    "Comment" : "Install a simple PHP application",
    "AWS::CloudFormation::Init" : {
      "config" : {
        "packages" : {
          "yum" : {
            "httpd"      : [],
            "php"         : []
          }
        },
        "files" : {

          "/var/www/html/index.php" : {
            "content" : { "Fn::Join" : ["", [
              "<?php\n",
              "echo '<h1>AWS CloudFormation sample PHP application</h1>';\n",
              "echo 'Updated version via UpdateStack';\n ",
              "?>\n"
            ]]},
            "mode"      : "000644",
            "owner"     : "apache",
            "group"     : "apache"
          },

          "/etc/cfn/cfn-hup.conf" : {
            "content" : { "Fn::Join" : ["", [
              "[main]\n",
              "stack=", { "Ref" : "AWS::StackId" }, "\n",
              "region=", { "Ref" : "AWS::Region" }, "\n"
            ]]},
            "mode"      : "000400",
            "owner"     : "root",
            "group"     : "root"
          },

          "/etc/cfn/hooks.d/cfn-auto-reloader.conf" : {
            "content": { "Fn::Join" : ["", [
              "[cfn-auto-reloader-hook]\n",

```

```

        "triggers=post.update\n",
        "path=Resources.LaunchConfig.Metadata.AWS::CloudFormation::Init\n",
        "action=/opt/aws/bin/cfn-init -s ", { "Ref" : "AWS::StackId" }, " -r
LaunchConfig ",
                                " --region      ", { "Ref" :
"AWS::Region" }, "\n",
        "runas=root\n"
    ]}]
    }
},

    "services" : {
        "sysvinit" : {
            "httpd"      : { "enabled" : "true", "ensureRunning" : "true" },
            "cfn-hup"    : { "enabled" : "true", "ensureRunning" : "true",
                "files" : ["/etc/cfn/cfn-hup.conf", "/etc/cfn/hooks.d/cfn-auto-
reloader.conf"]}]
        }
    }
}
},

    "Properties": {
        "ImageId" : { "Fn::FindInMap" : [ "AWSRegionArch2AMI", { "Ref" :
"AWS::Region" },
                                { "Fn::FindInMap" : [ "AWSInstanceType2Arch", { "Ref" :
"InstanceType" }, "Arch" ] } ] },
        "InstanceType" : { "Ref" : "InstanceType" },
        "KeyName"      : { "Ref" : "KeyName" },
        "SecurityGroups" : [ {"Ref" : "WebServerSecurityGroup"} ],
        "UserData"      : { "Fn::Base64" : { "Fn::Join" : [""], [
            "#!/bin/bash -xe\n",
            "yum install -y aws-cfn-bootstrap\n",

            "# Install the files and packages from the metadata\n",
            "/opt/aws/bin/cfn-init -v ",
            "      --stack ", { "Ref" : "AWS::StackName" },
            "      --resource LaunchConfig ",
            "      --region ", { "Ref" : "AWS::Region" }, "\n",

            "# Start up the cfn-hup daemon to listen for changes to the Web Server
metadata\n",
            "/opt/aws/bin/cfn-hup || error_exit 'Failed to start cfn-hup'\n",

```

```

        "# Signal the status from cfn-init\n",
        "/opt/aws/bin/cfn-signal -e $? ",
        "        --stack ", { "Ref" : "AWS::StackName" },
        "        --resource WebServerGroup ",
        "        --region ", { "Ref" : "AWS::Region" }, "\n"
    ]]]}
}
},

"WebServerSecurityGroup" : {
    "Type" : "AWS::EC2::SecurityGroup",
    "Properties" : {
        "GroupDescription" : "Enable HTTP access via port 80 locked down to the ELB and
SSH access",
        "SecurityGroupIngress" : [
            { "IpProtocol" : "tcp", "FromPort" : "80", "ToPort" : "80",
"SourceSecurityGroupOwnerId" : { "Fn::GetAtt" : [ "ElasticLoadBalancer",
"SourceSecurityGroup.OwnerAlias" ] }, "SourceSecurityGroupName" : { "Fn::GetAtt" :
[ "ElasticLoadBalancer", "SourceSecurityGroup.GroupName" ] } },
            { "IpProtocol" : "tcp", "FromPort" : "22", "ToPort" : "22", "CidrIp" :
{ "Ref" : "SSHLocation" } }
        ]
    }
},

"Outputs" : {
    "WebsiteURL" : {
        "Description" : "Application URL",
        "Value" : { "Fn::Join" : [ "", [ "http://", { "Fn::GetAtt" :
[ "ElasticLoadBalancer", "DNSName" ] } ] ] }
    }
}
}

```

Pertimbangan ketersediaan dan dampak

Properti yang berbeda memiliki dampak yang berbeda pada sumber daya dalam tumpukan. Anda dapat menggunakan CloudFormation untuk memperbarui properti apa pun; tetapi, sebelum Anda membuat perubahan, Anda harus mempertimbangkan pertanyaan-pertanyaan berikut:

1. Bagaimana pembaruan memengaruhi sumber daya itu sendiri? Misalnya, memperbarui ambang batas alarm akan membuat alarm tidak aktif selama pembaruan. Seperti yang telah kita lihat, mengubah jenis instance mengharuskan instance dihentikan dan dimulai ulang. CloudFormation menggunakan tindakan pembaruan atau modifikasi untuk sumber daya yang mendasarinya untuk membuat perubahan pada sumber daya. Untuk memahami dampak pembaruan, Anda harus memeriksa dokumentasi untuk sumber daya tertentu.
2. Apakah perubahan itu dapat berubah atau tidak dapat berubah? Beberapa perubahan pada properti sumber daya, seperti mengubah AMI di EC2 instans Amazon, tidak didukung oleh layanan yang mendasarinya. Dalam kasus perubahan yang bisa berubah, CloudFormation akan menggunakan jenis Update atau Modify APIs untuk sumber daya yang mendasarinya. Untuk perubahan properti yang tidak dapat diubah, CloudFormation akan membuat sumber daya baru dengan properti yang diperbarui dan kemudian menautkannya ke tumpukan sebelum menghapus sumber daya lama. Meskipun CloudFormation mencoba mengurangi waktu henti sumber daya tumpukan, mengganti sumber daya adalah proses multistep, dan itu akan memakan waktu. Selama konfigurasi ulang tumpukan, aplikasi Anda tidak akan beroperasi penuh. Misalnya, aplikasi mungkin tidak dapat melayani permintaan atau mengakses database.

Sumber daya terkait

Untuk informasi selengkapnya tentang penggunaan CloudFormation untuk memulai aplikasi dan mengintegrasikan dengan layanan konfigurasi dan penyebaran lainnya seperti Puppet dan Opscode Chef, lihat whitepaper berikut:

- [Bootstrapping aplikasi melalui CloudFormation](#)
- [Integrasi CloudFormation dengan Opscode Chef](#)
- [Integrasi CloudFormation dengan Wayang](#)

Template yang digunakan di seluruh bagian ini adalah aplikasi PHP “Hello World”. Pustaka template juga memiliki template ElastiCache sampel Amazon yang menunjukkan cara mengintegrasikan aplikasi PHP dengan ElastiCache menggunakan cfn-hup dan cfn-init untuk menanggapi perubahan dalam konfigurasi ElastiCache Cache Cluster Amazon, yang semuanya dapat dilakukan oleh Update Stack.

Lakukan blue/green penyebaran ECS melalui penggunaan CodeDeploy CloudFormation

Untuk memperbarui aplikasi yang berjalan di Amazon Elastic Container Service (Amazon ECS), Anda dapat menggunakan CodeDeploy strategi penyebaran biru/hijau. Strategi ini membantu meminimalkan gangguan yang disebabkan oleh perubahan versi aplikasi.

Dalam blue/green penerapan, Anda membuat lingkungan aplikasi baru (disebut hijau) di samping lingkungan hidup Anda saat ini (disebut sebagai biru). Ini memungkinkan Anda untuk memantau dan menguji lingkungan hijau sebelum merutekan lalu lintas langsung dari lingkungan biru ke lingkungan hijau. Setelah lingkungan hijau melayani lalu lintas langsung, Anda dapat dengan aman menghentikan lingkungan biru.

Untuk melakukan penerapan CodeDeploy biru/hijau pada ECS menggunakan CloudFormation, Anda menyertakan informasi berikut dalam template tumpukan Anda:

- `Hooks` Bagian yang menggambarkan `AWS::CodeDeploy::BlueGreen` kail.
- `Transform` Bagian yang menentukan `AWS::CodeDeployBlueGreen` transformasi.

Topik berikut memandu Anda melalui pengaturan CloudFormation template untuk penyebaran biru/hijau di ECS.

Topik

- [Tentang blue/green penerapan](#)
- [Pertimbangan saat mengelola penerapan biru/hijau ECS menggunakan CloudFormation](#)
- [AWS::CodeDeploy::BlueGreensintaks hook](#)
- [Contoh template penerapan biru/hijau](#)

Tentang blue/green penerapan

Topik ini memberikan ikhtisar tentang cara melakukan blue/green penerapan dengan CloudFormation pekerjaan. Ini juga menjelaskan bagaimana mempersiapkan CloudFormation template Anda untuk penerapan biru/hijau.

Topik

- [Cara kerjanya](#)

- [Pembaruan sumber daya yang memulai penerapan hijau](#)
- [Mempersiapkan templat Anda untuk melakukan deployment ECS biru/hijau](#)
- [Memodelkan blue/green penerapan Anda menggunakan sumber daya CloudFormation](#)
- [Ubah set](#)
- [Memantau peristiwa tumpukan](#)
- [Izin IAM untuk penerapan blue/green](#)

Cara kerjanya

Saat menggunakan CloudFormation untuk melakukan blue/green penerapan ECS CodeDeploy, Anda mulai dengan membuat template tumpukan yang menentukan sumber daya untuk lingkungan aplikasi biru dan hijau Anda, termasuk menentukan pengaturan perutean lalu lintas dan stabilisasi yang akan digunakan. Selanjutnya, Anda membuat tumpukan dari template itu. Ini menghasilkan aplikasi biru (saat ini) Anda. CloudFormation hanya membuat sumber daya biru selama pembuatan tumpukan. Sumber daya untuk deployment hijau tidak dibuat hingga diperlukan.

Kemudian, jika di update stack future Anda memperbarui definisi tugas atau sumber daya set tugas dalam aplikasi biru Anda, CloudFormation lakukan hal berikut:

- Menghasilkan semua sumber daya lingkungan aplikasi hijau yang diperlukan
- Menggeser lalu lintas berdasarkan parameter perutean lalu lintas yang ditentukan
- Menghapus sumber daya biru

Jika kesalahan terjadi pada titik mana pun sebelum deployment hijau berhasil dan diselesaikan, CloudFormation menggulung tumpukan kembali ke statusnya sebelum seluruh deployment hijau dimulai.

Pembaruan sumber daya yang memulai penerapan hijau

Saat Anda melakukan pembaruan tumpukan yang memperbarui properti tertentu dari sumber daya ECS tertentu, CloudFormation memulai proses penerapan hijau. Sumber daya yang memulai proses ini adalah:

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

Namun, jika pembaruan sumber daya ini tidak melibatkan perubahan properti yang memerlukan penggantian, penerapan hijau tidak akan dimulai. Untuk informasi selengkapnya, lihat [Memahami perilaku pembaruan sumber daya tumpukan](#).

Penting untuk dicatat bahwa Anda tidak dapat menggabungkan pembaruan ke sumber daya di atas dengan pembaruan ke sumber daya lain dalam operasi pembaruan tumpukan yang sama. Jika Anda perlu memperbarui sumber daya yang terdaftar dan sumber daya lain dalam tumpukan yang sama, Anda memiliki dua opsi:

- Lakukan dua operasi pembaruan tumpukan terpisah: satu yang hanya mencakup pembaruan ke sumber daya di atas, dan pembaruan tumpukan terpisah yang mencakup perubahan pada sumber daya lainnya.
- Hapus Transform dan Hooks bagian dari templat Anda dan kemudian lakukan pembaruan tumpukan. Dalam hal ini, CloudFormation tidak akan melakukan penerapan hijau.

Mempersiapkan templat Anda untuk melakukan deployment ECS biru/hijau

Untuk mengaktifkan blue/green penerapan pada tumpukan Anda, sertakan bagian berikut dalam template tumpukan Anda sebelum melakukan pembaruan tumpukan.

- Tambahkan referensi ke `AWS::CodeDeployBlueGreen` ubah ke templat anda:

```
"Transform": [  
  "AWS::CodeDeployBlueGreen"  
],
```

- Tambahkan Hooks bagian yang menjalankan `AWS::CodeDeploy::BlueGreen` dan menentukan properti untuk deployment Anda. Untuk informasi selengkapnya, lihat [AWS::CodeDeploy::BlueGreensintaks hook](#).
- Di bagian Resources, tentukan sumber daya biru dan hijau untuk deployment Anda.

Anda dapat menambahkan bagian ini ketika pertama kali membuat templat (yaitu, sebelum membuat tumpukan itu sendiri), atau Anda dapat menambahkannya ke templat yang ada sebelum melakukan pembaruan tumpukan. Jika Anda menentukan blue/green penerapan untuk tumpukan baru, CloudFormation hanya membuat sumber daya biru selama pembuatan tumpukan — sumber daya untuk penerapan hijau tidak dibuat sampai diperlukan selama pembaruan tumpukan.

Memodelkan blue/green penerapan Anda menggunakan sumber daya CloudFormation

Untuk melakukan penerapan CodeDeploy biru/hijau di ECS, CloudFormation template Anda harus menyertakan sumber daya yang memodelkan penerapan Anda, seperti layanan Amazon ECS dan penyeimbang beban. Untuk rincian lebih lanjut tentang apa yang diwakili sumber daya ini, lihat [Sebelum Anda memulai deployment Amazon ECS](#) dalam AWS CodeDeploy Panduan Pengguna.

Persyaratan	Sumber Daya	Diperlukan/Opsional	Memulai blue/green penerapan jika diganti?
Kluster Amazon ECS	AWS::ECS::Cluster	Opsional. Kluster default dapat digunakan.	Tidak
Layanan Amazon ECS	AWS::ECS::Service	Wajib.	Tidak
Penyeimbang Beban Jaringan atau Aplikasi	AWS::ECS::ServiceLoadBalancer	Wajib.	Tidak
Listener produksi	AWS::ElasticLoadBalancingV2::Listener	Wajib.	Tidak
Uji listener	AWS::ElasticLoadBalancingV2::Listener	Opsional.	Tidak
Dua grup target	AWS::ElasticLoadBalancingV2::TargetGroup	Wajib.	Tidak
Definisi tugas Amazon ECS	AWS::ECS::TaskDefinition	Wajib.	Ya
Wadah untuk aplikasi Amazon ECS Anda	AWS::ECS::TaskDefinitionContainerDefinitionName	Wajib.	Tidak

Persyaratan	Sumber Daya	Diperlukan/Opsional	Memulai blue/green penerapan jika diganti?
Port untuk set tugas pengganti Anda	AWS::ECS::TaskDefinition PortMapping ContainerPort	Wajib.	Tidak

Ubah set

Kami sangat menyarankan Anda untuk membuat perubahan yang ditetapkan sebelum melakukan pembaruan tumpukan yang akan memulai deployment hijau. Ini memungkinkan Anda untuk melihat perubahan aktual yang akan dilakukan pada tumpukan Anda sebelum melakukan pembaruan tumpukan. Sadarilah bahwa perubahan sumber daya mungkin tidak tercantum dalam urutan yang akan dilakukan selama pembaruan tumpukan. Untuk informasi selengkapnya, lihat [Perbarui CloudFormation tumpukan menggunakan set perubahan](#).

Memantau peristiwa tumpukan

Anda dapat melihat peristiwa tumpukan yang dihasilkan pada setiap langkah deployment ECS pada tab Peristiwa dari halaman Tumpukan, dan menggunakan AWS CLI. Untuk informasi selengkapnya, lihat [Pantau kemajuan tumpukan](#).

Izin IAM untuk penerapan blue/green

Agar berhasil CloudFormation melakukan penerapan biru-hijau, Anda harus memiliki izin berikut: CodeDeploy

- `codedeploy:Get*`
- `codedeploy:CreateCloudFormationDeployment`

Untuk informasi selengkapnya, lihat [Kunci tindakan, sumber daya, dan kondisi CodeDeploy](#) di Referensi Otorisasi Layanan.

Pertimbangan saat mengelola penerapan biru/hijau ECS menggunakan CloudFormation

Proses penggunaan CloudFormation untuk melakukan penyebaran ECS Anda berbeda dari blue/green penerapan CodeDeploy ECS standar yang hanya menggunakan CodeDeploy. Untuk pemahaman mendetail tentang perbedaan-perbedaan ini, lihat [Perbedaan antara blue/green penerapan Amazon ECS melalui CodeDeploy dan AWS CloudFormation di Panduan Pengguna AWS CodeDeploy](#).

Saat mengelola blue/green penerapan Anda menggunakan CloudFormation, ada batasan dan pertimbangan tertentu yang perlu diingat:

- Hanya pembaruan ke sumber daya tertentu yang akan memulai penerapan hijau. Untuk informasi selengkapnya, lihat [Pembaruan sumber daya yang memulai penerapan hijau](#).
- Anda tidak dapat menyertakan pembaruan ke sumber daya yang memulai penerapan hijau dan pembaruan ke sumber daya lain dalam pembaruan tumpukan yang sama. Untuk informasi selengkapnya, lihat [Pembaruan sumber daya yang memulai penerapan hijau](#).
- Anda hanya dapat menentukan layanan ECS tunggal sebagai target deployment.
- Parameter yang nilainya dikaburkan oleh CloudFormation dapat diperbarui CodeDeploy selama penerapan hijau, dan akan menyebabkan kesalahan dan kegagalan pembaruan tumpukan. Ini termasuk:
 - Parameter didefinisikan dengan NoEcho atribut.
 - Parameter yang menggunakan referensi dinamis untuk mengambil nilai-nilainya dari layanan eksternal. Untuk informasi selengkapnya tentang referensi dinamis, lihat [Dapatkan nilai yang disimpan di layanan lain menggunakan referensi dinamis](#).
- Untuk membatalkan penerapan hijau yang masih berlangsung, batalkan pembaruan tumpukan di CloudFormation, bukan CodeDeploy atau ECS. Untuk informasi selengkapnya, lihat [Batalkan pembaruan tumpukan](#). Setelah pembaruan selesai, Anda tidak dapat membatalkannya. Namun, Anda dapat memperbarui tumpukan lagi dengan pengaturan sebelumnya.
- CloudFormation Fitur-fitur berikut saat ini tidak didukung untuk template yang menentukan blue/green penerapan ECS:
 - Mendeklarasikan [output](#) atau menggunakan [Fn:: ImportValue](#) untuk mengimpor nilai dari tumpukan lain.
 - Mengimpor sumber daya. Untuk informasi selengkapnya tentang mengimpor sumber daya, lihat [Impor AWS sumber daya ke CloudFormation tumpukan](#).

- Menggunakan `AWS::CodeDeploy::BlueGreen` hook dalam template yang menyertakan sumber daya tumpukan bersarang. Untuk informasi lebih lanjut tentang tumpukan bersarang, lihat [Pisahkan template menjadi potongan-potongan yang dapat digunakan kembali menggunakan tumpukan bersarang](#)
- Menggunakan `AWS::CodeDeploy::BlueGreen` hook di tumpukan bersarang.

AWS::CodeDeploy::BlueGreensintaks hook

Sintaks berikut menjelaskan struktur `AWS::CodeDeploy::BlueGreen` hook untuk penyebaran ECS. blue/green

Sintaksis

```
"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)"
      ]
    }
  }
}

```

Properti

Logical ID (juga disebut nama logis)

ID logis dari hook dideklarasikan di Hooks bagian template. ID logis harus berupa alfanumerik (A-Za-z0-9) dan unik dalam templat.

Wajib: Ya

Type

Jenis kait. `AWS::CodeDeploy::BlueGreen`

Wajib: Ya

Properties

Properti kait.

Wajib: Ya

`TrafficRoutingConfig`

Pengaturan konfigurasi perutean lalu lintas.

Wajib: Tidak

Konfigurasi default adalah pergeseran lalu lintas Canary berbasis waktu, dengan persentase langkah 15% dan waktu menanam lima menit.

Type

Jenis pergeseran lalu lintas yang digunakan oleh konfigurasi deployment.

Nilai yang valid: `AllAtOnce` | `TimeBasedCanary` | `TimeBasedLinear`

Wajib: Ya

`TimeBasedCanary`

Menentukan konfigurasi yang menggeser lalu lintas dari satu versi deployment ke versi lain dalam dua tambahan.

Wajib: Bersyarat: Jika Anda menentukan `TimeBasedCanary` sebagai jenis perutean lalu lintas, Anda harus menyertakan parameter. `TimeBasedCanary`

`StepPercentage`

Persentase lalu lintas bergeser dalam tambahan pertama `TimeBasedCanary` deployment. Persentase langkah harus 14% atau lebih besar.

Wajib: Tidak

`BakeTimeMins`

Jumlah menit antara pergeseran lalu lintas pertama dan kedua dari `TimeBasedCanary` deployment.

Wajib: Tidak

TimeBasedLinear

Menentukan konfigurasi yang menggeser lalu lintas dari satu versi deployment ke versi lain dalam dua tambahan, dengan jumlah menit yang sama antara setiap tambahan.

Wajib: Bersyarat: Jika Anda menentukan TimeBasedLinear sebagai jenis perutean lalu lintas, Anda harus menyertakan parameter. TimeBasedLinear

StepPercentage

Persentase lalu lintas yang bergeser pada awal setiap tambahan TimeBasedLinear deployment. Persentase langkah harus 14% atau lebih besar.

Wajib: Tidak

BakeTimeMins

Jumlah menit antara setiap pergeseran lalu lintas tambahan TimeBasedLinear deployment.

Wajib: Tidak

AdditionalOptions

Opsi tambahan untuk blue/green penyebaran.

Wajib: Tidak

TerminationWaitTimeInMinutes

Menentukan waktu tunggu, dalam hitungan menit, sebelum mengakhiri sumber daya biru.

Wajib: Tidak

LifecycleEventHooks

Gunakan kait peristiwa siklus hidup untuk menentukan fungsi Lambda yang CodeDeploy dapat memanggil untuk memvalidasi penerapan. Anda dapat menggunakan fungsi yang sama atau yang berbeda untuk peristiwa deployment siklus hidup. Setelah menyelesaikan tes validasi, fungsi AfterAllowTraffic Lambda memanggil CodeDeploy kembali dan

memberikan hasil dari atau. Succeeded Failed Untuk informasi selengkapnya, lihat [bagian AppSpec 'kait'](#) di Panduan AWS CodeDeploy Pengguna.

Wajib: Tidak

`BeforeInstall`

Fungsi yang digunakan untuk menjalankan tugas sebelum set tugas pengganti dibuat.

Wajib: Tidak

`AfterInstall`

Fungsi yang digunakan untuk menjalankan tugas setelah set tugas pengganti dibuat dan salah satu grup target terkait dengan itu.

Wajib: Tidak

`AfterAllowTestTraffic`

Fungsi yang digunakan untuk menjalankan tugas setelah listener tes melayani lalu lintas ke set tugas pengganti.

Wajib: Tidak

`BeforeAllowTraffic`

Fungsi yang digunakan untuk menjalankan tugas setelah grup target kedua dikaitkan dengan set tugas pengganti, tapi sebelum lalu lintas bergeser ke set tugas pengganti.

Wajib: Tidak

`AfterAllowTraffic`

Fungsi yang digunakan untuk menjalankan tugas setelah grup target kedua melayani lalu lintas ke set tugas pengganti.

Wajib: Tidak

`ServiceRole`

Peran eksekusi untuk digunakan CloudFormation untuk melakukan penyebaran biru-hijau. Untuk daftar izin yang diperlukan, lihat [Izin IAM untuk penerapan blue/green](#).

Wajib: Tidak

`Applications`

Menentukan properti aplikasi Amazon ECS.

Wajib: Ya

Target

Wajib: Ya

Type

Jenis sumber daya.

Wajib: Ya

LogicalID

ID logis sumber daya.

Wajib: Ya

ECSAttributes

Sumber daya yang mewakili berbagai persyaratan deployment aplikasi Amazon ECS Anda.

Wajib: Ya

TaskDefinitions

ID logis [AWS::ECS::TaskDefinition](#) sumber daya untuk menjalankan kontainer Docker yang berisi aplikasi Amazon ECS Anda.

Wajib: Ya

TaskSets

Logika IDs sumber [AWS::ECS::TaskSet](#) daya untuk digunakan sebagai set tugas untuk aplikasi.

Wajib: Ya

TrafficRouting

Menentukan sumber daya yang digunakan untuk perutean lalu lintas.

Wajib: Ya

ProdTrafficRoute

Listener yang akan digunakan oleh penyeimbangan beban Anda untuk mengarahkan lalu lintas ke grup target Anda.

Wajib: Ya

Type

Jenis sumber daya. `AWS::ElasticLoadBalancingV2::Listener`

Wajib: Ya

LogicalID

ID logis sumber daya.

Wajib: Ya

TestTrafficRoute

Listener yang akan digunakan oleh penyeimbangan beban Anda untuk mengarahkan lalu lintas ke grup target Anda.

Wajib: Ya

Type

Jenis sumber daya. `AWS::ElasticLoadBalancingV2::Listener`

Wajib: Ya

LogicalID

ID logis sumber daya.

Wajib: Tidak

TargetGroups

ID logis sumber daya yang digunakan sebagai grup target untuk merutekan lalu lintas ke target terdaftar.

Wajib: Ya

Contoh template penerapan biru/hijau

Contoh template berikut menyiapkan penyebaran CodeDeploy biru/hijau di ECS, dengan kemajuan perutean lalu lintas 15 persen per langkah dan periode stabilisasi 5 menit antara setiap langkah.

Membuat tumpukan dengan template akan menyediakan konfigurasi awal penerapan. Jika Anda kemudian membuat perubahan pada properti di `BlueTaskSet` sumber daya yang mengharuskan

sumber daya diganti, maka CloudFormation akan memulai penerapan hijau sebagai bagian dari pembaruan tumpukan.

JSON

```
{
  "AWSTemplateFormatVersion":"2010-09-09",
  "Parameters":{
    "Vpc":{"Type":"AWS::EC2::VPC::Id" },
    "Subnet1":{"Type":"AWS::EC2::Subnet::Id" },
    "Subnet2":{"Type":"AWS::EC2::Subnet::Id" }
  },
  "Transform":[ "AWS::CodeDeployBlueGreen" ],
  "Hooks":{
    "CodeDeployBlueGreenHook":{
      "Type":"AWS::CodeDeploy::BlueGreen",
      "Properties":{
        "TrafficRoutingConfig":{
          "Type":"TimeBasedCanary",
          "TimeBasedCanary":{
            "StepPercentage":15,
            "BakeTimeMins":5
          }
        }
      },
    },
    "Applications":[
      {
        "Target":{
          "Type":"AWS::ECS::Service",
          "LogicalID":"ECSDemoService"
        },
        "ECSAttributes":{
          "TaskDefinitions":[ "BlueTaskDefinition","GreenTaskDefinition" ],
          "TaskSets":[ "BlueTaskSet","GreenTaskSet" ],
          "TrafficRouting":{
            "ProdTrafficRoute":{
              "Type":"AWS::ElasticLoadBalancingV2::Listener",
              "LogicalID":"ALBListenerProdTraffic"
            },
            "TargetGroups":[ "ALBTargetGroupBlue","ALBTargetGroupGreen" ]
          }
        }
      }
    ]
  }
}
```

```
    }
  },
  "Resources":{
    "ExampleSecurityGroup":{
      "Type":"AWS::EC2::SecurityGroup",
      "Properties":{
        "GroupDescription":"Security group for ec2 access",
        "VpcId":{"Ref":"Vpc" },
        "SecurityGroupIngress":[
          {
            "IpProtocol":"tcp",
            "FromPort":80,
            "ToPort":80,
            "CidrIp":"0.0.0.0/0"
          },
          {
            "IpProtocol":"tcp",
            "FromPort":8080,
            "ToPort":8080,
            "CidrIp":"0.0.0.0/0"
          },
          {
            "IpProtocol":"tcp",
            "FromPort":22,
            "ToPort":22,
            "CidrIp":"0.0.0.0/0"
          }
        ]
      }
    }
  },
  "ALBTargetGroupBlue":{
    "Type":"AWS::ElasticLoadBalancingV2::TargetGroup",
    "Properties":{
      "HealthCheckIntervalSeconds":5,
      "HealthCheckPath":"/",
      "HealthCheckPort":"80",
      "HealthCheckProtocol":"HTTP",
      "HealthCheckTimeoutSeconds":2,
      "HealthyThresholdCount":2,
      "Matcher":{"HttpCode":"200" },
      "Port":80,
      "Protocol":"HTTP",
      "Tags":[{"Key":"Group","Value":"Example" }],
      "TargetType":"ip",
```

```

        "UnhealthyThresholdCount":4,
        "VpcId":{"Ref":"Vpc" }
    }
},
"ALBTargetGroupGreen":{"
    "Type":"AWS::ElasticLoadBalancingV2::TargetGroup",
    "Properties":{"
        "HealthCheckIntervalSeconds":5,
        "HealthCheckPath":"/",
        "HealthCheckPort":"80",
        "HealthCheckProtocol":"HTTP",
        "HealthCheckTimeoutSeconds":2,
        "HealthyThresholdCount":2,
        "Matcher":{"HttpCode":"200" },
        "Port":80,
        "Protocol":"HTTP",
        "Tags":[{"Key":"Group","Value":"Example" }],
        "TargetType":"ip",
        "UnhealthyThresholdCount":4,
        "VpcId":{"Ref":"Vpc" }
    }
},
"ExampleALB":{"
    "Type":"AWS::ElasticLoadBalancingV2::LoadBalancer",
    "Properties":{"
        "Scheme":"internet-facing",
        "SecurityGroups":[{"Ref":"ExampleSecurityGroup" }],
        "Subnets":[{"Ref":"Subnet1" },{ "Ref":"Subnet2" }],
        "Tags":[{"Key":"Group","Value":"Example" }],
        "Type":"application",
        "IpAddressType":"ipv4"
    }
},
"ALBListenerProdTraffic":{"
    "Type":"AWS::ElasticLoadBalancingV2::Listener",
    "Properties":{"
        "DefaultActions":[
            {
                "Type":"forward",
                "ForwardConfig":{"
                    "TargetGroups":[
                        {
                            "TargetGroupArn":{"Ref":"ALBTargetGroupBlue" },
                            "Weight":1

```

```

        }
      ]
    }
  ],
  "LoadBalancerArn":{ "Ref":"ExampleALB" },
  "Port":80,
  "Protocol":"HTTP"
}
},
"ALBListenerProdRule":{
  "Type":"AWS::ElasticLoadBalancingV2::ListenerRule",
  "Properties":{
    "Actions":[
      {
        "Type":"forward",
        "ForwardConfig":{
          "TargetGroups":[
            {
              "TargetGroupArn":{ "Ref":"ALBTargetGroupBlue" },
              "Weight":1
            }
          ]
        }
      ]
    },
    "Conditions":[
      {
        "Field":"http-header",
        "HttpHeaderConfig":{
          "HttpHeaderName":"User-Agent",
          "Values":[ "Mozilla" ]
        }
      }
    ]
  },
  "ListenerArn":{ "Ref":"ALBListenerProdTraffic" },
  "Priority":1
}
},
"ECSTaskExecutionRole":{
  "Type":"AWS::IAM::Role",
  "Properties":{
    "AssumeRolePolicyDocument":{
      "Version":"2012-10-17",

```



```

        "Statement": [
            {
                "Sid": "",
                "Effect": "Allow",
                "Principal": {
                    "Service": "ecs-tasks.amazonaws.com"
                },
                "Action": "sts:AssumeRole"
            }
        ],
        "ManagedPolicyArns": [ "arn:aws:iam::aws:policy/service-role/AmazonECSTaskExecutionRolePolicy" ]
    },
    "BlueTaskDefinition": {
        "Type": "AWS::ECS::TaskDefinition",
        "Properties": {
            "ExecutionRoleArn": {
                "Fn::GetAtt": [ "ECSTaskExecutionRole", "Arn" ]
            },
            "ContainerDefinitions": [
                {
                    "Name": "DemoApp",
                    "Image": "nginxdemos/hello:latest",
                    "Essential": true,
                    "PortMappings": [
                        {
                            "HostPort": 80,
                            "Protocol": "tcp",
                            "ContainerPort": 80
                        }
                    ]
                }
            ]
        }
    },
    "RequiresCompatibilities": [ "FARGATE" ],
    "NetworkMode": "awsvpc",
    "Cpu": "256",
    "Memory": "512",
    "Family": "ecs-demo"
},
"ECSDemoCluster": {
    "Type": "AWS::ECS::Cluster",

```

```

    "Properties":{}
  },
  "ECSDemoService":{
    "Type":"AWS::ECS::Service",
    "Properties":{
      "Cluster":{ "Ref":"ECSDemoCluster" },
      "DesiredCount":1,
      "DeploymentController":{ "Type":"EXTERNAL" }
    }
  },
  "BlueTaskSet":{
    "Type":"AWS::ECS::TaskSet",
    "Properties":{
      "Cluster":{ "Ref":"ECSDemoCluster" },
      "LaunchType":"FARGATE",
      "NetworkConfiguration":{
        "AwsVpcConfiguration":{
          "AssignPublicIp":"ENABLED",
          "SecurityGroups":[{ "Ref":"ExampleSecurityGroup" }],
          "Subnets":[{ "Ref":"Subnet1" },{ "Ref":"Subnet2" }]
        }
      },
      "PlatformVersion":"1.4.0",
      "Scale":{
        "Unit":"PERCENT",
        "Value":100
      },
      "Service":{ "Ref":"ECSDemoService"},
      "TaskDefinition":{ "Ref":"BlueTaskDefinition" },
      "LoadBalancers":[
        {
          "ContainerName":"DemoApp",
          "ContainerPort":80,
          "TargetGroupArn":{ "Ref":"ALBTargetGroupBlue" }
        }
      ]
    }
  },
  "PrimaryTaskSet":{
    "Type":"AWS::ECS::PrimaryTaskSet",
    "Properties":{
      "Cluster":{ "Ref":"ECSDemoCluster" },
      "Service":{ "Ref":"ECSDemoService" },
      "TaskSetId":{ "Fn::GetAtt":[ "BlueTaskSet","Id" ]

```

```

    }
  }
}
}
}

```

YAML

```

AWSTemplateFormatVersion: 2010-09-09
Parameters:
  Vpc:
    Type: 'AWS::EC2::VPC::Id'
  Subnet1:
    Type: 'AWS::EC2::Subnet::Id'
  Subnet2:
    Type: 'AWS::EC2::Subnet::Id'
Transform:
  - 'AWS::CodeDeployBlueGreen'
Hooks:
  CodeDeployBlueGreenHook:
    Type: 'AWS::CodeDeploy::BlueGreen'
    Properties:
      TrafficRoutingConfig:
        Type: TimeBasedCanary
        TimeBasedCanary:
          StepPercentage: 15
          BakeTimeMins: 5
      Applications:
        - Target:
            Type: 'AWS::ECS::Service'
            LogicalID: ECSDemoService
          ECSAttributes:
            TaskDefinitions:
              - BlueTaskDefinition
              - GreenTaskDefinition
            TaskSets:
              - BlueTaskSet
              - GreenTaskSet
            TrafficRouting:
              ProdTrafficRoute:
                Type: 'AWS::ElasticLoadBalancingV2::Listener'
                LogicalID: ALBListenerProdTraffic
              TargetGroups:

```

- ALBTargetGroupBlue
- ALBTargetGroupGreen

Resources:**ExampleSecurityGroup:**

Type: 'AWS::EC2::SecurityGroup'

Properties:

GroupDescription: Security group for ec2 access

VpcId: !Ref Vpc

SecurityGroupIngress:

- IpProtocol: tcp
FromPort: 80
ToPort: 80
CidrIp: 0.0.0.0/0
- IpProtocol: tcp
FromPort: 8080
ToPort: 8080
CidrIp: 0.0.0.0/0
- IpProtocol: tcp
FromPort: 22
ToPort: 22
CidrIp: 0.0.0.0/0

ALBTargetGroupBlue:

Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'

Properties:

HealthCheckIntervalSeconds: 5

HealthCheckPath: /

HealthCheckPort: '80'

HealthCheckProtocol: HTTP

HealthCheckTimeoutSeconds: 2

HealthyThresholdCount: 2

Matcher:

HttpCode: '200'

Port: 80

Protocol: HTTP

Tags:

- Key: Group
Value: Example

TargetType: ip

UnhealthyThresholdCount: 4

VpcId: !Ref Vpc

ALBTargetGroupGreen:

Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'

Properties:

HealthCheckIntervalSeconds: 5

```
HealthCheckPath: /
HealthCheckPort: '80'
HealthCheckProtocol: HTTP
HealthCheckTimeoutSeconds: 2
HealthyThresholdCount: 2
Matcher:
  HttpCode: '200'
Port: 80
Protocol: HTTP
Tags:
  - Key: Group
    Value: Example
TargetType: ip
UnhealthyThresholdCount: 4
VpcId: !Ref Vpc
ExampleALB:
  Type: 'AWS::ElasticLoadBalancingV2::LoadBalancer'
  Properties:
    Scheme: internet-facing
    SecurityGroups:
      - !Ref ExampleSecurityGroup
    Subnets:
      - !Ref Subnet1
      - !Ref Subnet2
    Tags:
      - Key: Group
        Value: Example
    Type: application
    IpAddressType: ipv4
ALBListenerProdTraffic:
  Type: 'AWS::ElasticLoadBalancingV2::Listener'
  Properties:
    DefaultActions:
      - Type: forward
        ForwardConfig:
          TargetGroups:
            - TargetGroupArn: !Ref ALBTargetGroupBlue
              Weight: 1
    LoadBalancerArn: !Ref ExampleALB
    Port: 80
    Protocol: HTTP
ALBListenerProdRule:
  Type: 'AWS::ElasticLoadBalancingV2::ListenerRule'
  Properties:
```

```
Actions:
  - Type: forward
    ForwardConfig:
      TargetGroups:
        - TargetGroupArn: !Ref ALBTargetGroupBlue
          Weight: 1
Conditions:
  - Field: http-header
    HttpHeaderConfig:
      HttpHeadersName: 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
```

CloudFormation cuplikan template

Bagian ini menyediakan sejumlah contoh skenario yang dapat Anda gunakan untuk memahami cara menyatakan berbagai bagian templat AWS CloudFormation . Anda juga dapat menggunakan cuplikan sebagai titik awal untuk bagian templat kustom Anda.

Note

Karena AWS CloudFormation template harus sesuai dengan JSON, tidak ada ketentuan untuk karakter kelanjutan baris. Pembungkus cuplikan dalam dokumen ini mungkin acak jika garis lebih panjang dari 68 karakter.

Topik

- [Snippet templat umum](#)
- [Cuplikan CloudFormation templat penskalaan otomatis](#)
- [AWS Cuplikan templat Konsol Penagihan](#)
- [AWS CloudFormation cuplikan template](#)
- [Cuplikan CloudFront template Amazon](#)
- [Cuplikan CloudWatch template Amazon](#)
- [Cuplikan templat Amazon CloudWatch Log](#)
- [Cuplikan template Amazon DynamoDB](#)
- [Cuplikan EC2 CloudFormation template Amazon](#)
- [Templat contoh Layanan Kontainer Elastis Amazon](#)
- [Templat Contoh Sistem File Elastis Amazon](#)
- [Cuplikan templat Elastic Beanstalk](#)
- [Cuplikan templat Elastic Load Balancing](#)
- [AWS Identity and Access Management cuplikan template](#)
- [AWS Lambda Template](#)
- [AWS OpsWorks cuplikan template](#)
- [Cuplikan template Amazon Redshift](#)

- [Cuplikan template Amazon RDS](#)
- [Cuplikan templat Route 53](#)
- [Cuplikan template Amazon S3](#)
- [Cuplikan template Amazon SNS](#)
- [Cuplikan template Amazon SQS](#)
- [Cuplikan template Amazon Timestream](#)

Snippet templat umum

Contoh berikut menunjukkan fitur CloudFormation template berbeda yang tidak spesifik untuk AWS layanan.

Topik

- [Base64properti dikodekan UserData](#)
- [Base64UserDataproperti yang dikodekan dengan dan AccessKeySecretKey](#)
- [Parametersbagian dengan satu parameter string literal](#)
- [Parametersbagian dengan parameter string dengan kendala ekspresi reguler](#)
- [Parametersbagian dengan parameter angka dengan MinValue dan MaxValue kendala](#)
- [Parametersbagian dengan parameter angka dengan AllowedValues kendala](#)
- [Parametersbagian dengan satu CommaDelimitedList parameter literal](#)
- [Parametersbagian dengan nilai parameter berdasarkan parameter semu](#)
- [Mappingbagian dengan tiga pemetaan](#)
- [Descriptionberdasarkan string literal](#)
- [Outputsbagian dengan satu output string literal](#)
- [Outputsbagian dengan satu referensi sumber daya dan satu output referensi semu](#)
- [Outputsbagian dengan output berdasarkan fungsi, string literal, referensi, dan parameter semu](#)
- [Versi format templat](#)
- [AWSTagsproperti](#)

Base64properti dikodekan UserData

Contoh ini menunjukkan perakitan UserData properti menggunakan Fn::Base64 dan Fn::Join fungsi. Referensi MyValue dan MyName merupakan parameter yang harus ditentukan di

Parameters bagian template. String literal Hello World hanyalah nilai lain yang diberikan contoh ini sebagai bagian dari UserData.

JSON

```
"UserData" : {
  "Fn::Base64" : {
    "Fn::Join" : [ ",", [
      { "Ref" : "MyValue" },
      { "Ref" : "MyName" },
      "Hello World" ] ]
  }
}
```

YAML

```
UserData:
  Fn::Base64: !Sub |
    Ref: MyValue
    Ref: MyName
    Hello World
```

Base64UserData properti yang dikodekan dengan dan AccessKeySecretKey

Contoh ini menunjukkan perakitan UserData properti menggunakan Fn::Base64 dan Fn::Join fungsi. Ini menyertakan informasi AccessKey dan SecretKey. Referensi AccessKey dan SecretKey adalah parameter yang harus ditentukan di bagian Parameter dari templat.

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 bagian dengan satu parameter string literal

Contoh berikut menggambarkan pernyataan bagian Parameter yang valid di mana parameter tipe String tunggal dinyatakan.

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 bagian dengan parameter string dengan kendala ekspresi reguler

Contoh berikut menggambarkan pernyataan bagian Parameter yang valid di mana parameter tipe String tunggal dinyatakan. AdminUserAccountParameter memiliki default dari admin. Nilai parameter harus memiliki panjang minimum 1, panjang maksimum 16, dan berisi karakter abjad dan angka tetapi harus dimulai dengan karakter abjad.

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 bagian dengan parameter angka dengan MinValue dan MaxValue kendala

Contoh berikut menggambarkan pernyataan bagian Parameter yang valid di mana parameter tipe Number tunggal dinyatakan. WebServerPortParameter memiliki default 80 dan nilai minimum 1 dan nilai maksimum 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
```

Parameters bagian dengan parameter angka dengan AllowedValues kendala

Contoh berikut menggambarkan pernyataan bagian Parameter yang valid di mana parameter tipe Number tunggal dinyatakan. `WebServerPortParameter` memiliki default 80 dan hanya memungkinkan nilai 80 dan 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 bagian dengan satu CommaDelimitedList parameter literal

Contoh berikut menggambarkan deklarasi `Parameters` bagian yang valid di mana parameter `CommaDelimitedList` tipe tunggal dideklarasikan. `NoEchoProperti` diatur ke `TRUE`, yang akan menutupi nilainya dengan tanda bintang (*****) dalam `describe-stacks` output, kecuali untuk informasi yang disimpan di lokasi yang ditentukan di bawah ini.

⚠ Important

Menggunakan NoEcho atribut tidak menutupi informasi apa pun yang disimpan sebagai berikut:

- Bagian Metadata template. CloudFormation tidak mengubah, memodifikasi, atau menyunting informasi apa pun yang Anda sertakan di Metadata bagian ini. Untuk informasi selengkapnya, lihat [Metadata](#).
- Bagian Outputs template. Untuk informasi lebih lanjut, lihat [Output](#).
- MetadataAtribut definisi sumber daya. Untuk informasi lebih lanjut, lihat [Atribut Metadata](#).

Kami sangat menyarankan Anda untuk tidak menggunakan mekanisme ini untuk memasukkan informasi sensitif, seperti kata sandi atau rahasia.

⚠ Important

Daripada menyematkan informasi sensitif secara langsung di CloudFormation template Anda, kami sarankan Anda menggunakan parameter dinamis dalam template tumpukan untuk mereferensikan informasi sensitif yang disimpan dan dikelola di luar CloudFormation, seperti di AWS Systems Manager Parameter Store atau AWS Secrets Manager.

Untuk informasi selengkapnya, lihat [Do not embed credentials dalam praktik terbaik template Anda](#).

JSON

```
"Parameters" : {
  "UserRoles" : {
    "Type" : "CommaDelimitedList",
    "Default" : "guest,newhire",
    "NoEcho" : "TRUE"
  }
}
```

YAML

```
Parameters:
  UserRoles:
    Type: CommaDelimitedList
    Default: "guest,newhire"
    NoEcho: true
```

Parameters bagian dengan nilai parameter berdasarkan parameter semu

Contoh berikut menunjukkan perintah dalam data EC2 pengguna yang menggunakan parameter pseudo `AWS::StackName` dan `AWS::Region`. Untuk informasi selengkapnya tentang parameter semu, lihat [Dapatkan AWS nilai menggunakan parameter semu](#).

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}
```

Mapping bagian dengan tiga pemetaan

Contoh berikut menggambarkan deklarasi Mapping bagian yang valid yang berisi tiga pemetaan. Peta tersebut, ketika dicocokkan dengan kunci pemetaan Stop, SlowDown, atau Go, menyediakan nilai-nilai RGB yang ditetapkan untuk atribut RGBColor terkait.

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 berdasarkan string literal

Contoh berikut menggambarkan deklarasi Description bagian yang valid di mana nilai didasarkan pada string literal. Snippet ini bisa untuk templat, parameter, sumber daya, properti, atau output.

JSON

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

YAML

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

Outputs bagian dengan satu output string literal

Contoh ini menunjukkan tugas output berdasarkan string literal.

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 bagian dengan satu referensi sumber daya dan satu output referensi semu

Contoh ini menunjukkan Outputs bagian dengan dua tugas keluaran. Salah satunya didasarkan pada sumber daya, dan yang lainnya didasarkan pada referensi semu.

JSON

```
"Outputs" : {
```

```
"SNSTopic" : { "Value" : { "Ref" : "MyNotificationTopic" } },
"StackName" : { "Value" : { "Ref" : "AWS::StackName" } }
}
```

YAML

```
Outputs:
  SNSTopic:
    Value: !Ref MyNotificationTopic
  StackName:
    Value: !Ref AWS::StackName
```

Outputs bagian dengan output berdasarkan fungsi, string literal, referensi, dan parameter semu

Contoh ini menunjukkan bagian Output dengan satu tugas output. Fungsi Join digunakan untuk menggabungkan nilai, menggunakan tanda persen sebagai pembatas.

JSON

```
"Outputs" : {
  "MyOutput" : {
    "Value" : { "Fn::Join" :
      [ "%", [ "A-string", {"Ref" : "AWS::StackName" } ] ]
    }
  }
}
```

YAML

```
Outputs:
  MyOutput:
    Value: !Join [ %, [ 'A-string', !Ref 'AWS::StackName' ] ]
```

Versi format templat

Cuplikan berikut menggambarkan deklarasi bagian yang valid `AWSTemplateFormatVersion`.

JSON

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

YAML

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

AWSTagsproperty

Contoh ini menunjukkan AWS Tags property. Anda akan menentukan property ini di dalam bagian Property dari sumber daya. Ketika sumber daya dibuat, sumber daya tersebut akan ditandai dengan tanda yang Anda nyatakan.

JSON

```
"Tags" : [  
  {  
    "Key" : "keyname1",  
    "Value" : "value1"  
  },  
  {  
    "Key" : "keyname2",  
    "Value" : "value2"  
  }  
]
```

YAML

```
Tags:  
-  
  Key: "keyname1"  
  Value: "value1"  
-  
  Key: "keyname2"  
  Value: "value2"
```

Cuplikan CloudFormation templat penskalaan otomatis

Dengan Amazon EC2 Auto Scaling, Anda dapat secara otomatis menskalakan EC2 instans Amazon, baik dengan kebijakan penskalaan atau dengan penskalaan terjadwal. Grup Auto Scaling adalah kumpulan EC2 instans Amazon yang memungkinkan penskalaan otomatis dan fitur pengelolaan armada, seperti kebijakan penskalaan, tindakan terjadwal, pemeriksaan kesehatan, kait siklus hidup, dan penyeimbangan beban.

Application Auto Scaling menyediakan penskalaan otomatis berbagai sumber daya di luar Amazon EC2, baik dengan kebijakan penskalaan atau dengan penskalaan terjadwal.

Anda dapat membuat dan mengonfigurasi grup Auto Scaling, kebijakan penskalaan, tindakan terjadwal, dan sumber daya penskalaan otomatis lainnya sebagai bagian dari infrastruktur menggunakan templat. AWS CloudFormation Template memudahkan pengelolaan dan otomatisasi penyebaran sumber daya penskalaan otomatis secara berulang dan konsisten.

Contoh cuplikan template berikut menjelaskan AWS CloudFormation sumber daya atau komponen untuk Amazon Auto EC2 Scaling dan Application Auto Scaling. Cuplikan ini dirancang untuk diintegrasikan ke dalam template dan tidak dimaksudkan untuk dijalankan secara independen.

Kategori Snippet

- [Konfigurasi sumber EC2 daya Amazon Auto Scaling dengan AWS CloudFormation](#)
- [Konfigurasi sumber daya Application Auto Scaling dengan AWS CloudFormation](#)

Konfigurasi sumber EC2 daya Amazon Auto Scaling dengan AWS CloudFormation

Contoh berikut menunjukkan cuplikan berbeda untuk disertakan dalam templat untuk digunakan dengan Amazon Auto EC2 Scaling.

Kategori cuplikan

- [Buat satu contoh grup Auto Scaling](#)
- [Buat grup Auto Scaling dengan penyeimbang beban terlampir](#)
- [Buat grup Auto Scaling dengan notifikasi](#)
- [Buat grup Auto Scaling yang menggunakan a dan CreationPolicyUpdatePolicy](#)
- [Membuat kebijakan penskalaan langkah](#)
- [Contoh kelompok contoh campuran](#)
- [Luncurkan contoh konfigurasi](#)

Buat satu contoh grup Auto Scaling

Contoh ini menunjukkan [AWS::AutoScaling::AutoScalingGroup](#) sumber daya dengan satu instance untuk membantu Anda memulai. `VPCZoneIdentifier` Properti grup Auto Scaling menentukan daftar subnet yang ada di tiga Availability Zone yang berbeda. Anda harus menentukan subnet yang berlaku IDs dari akun Anda sebelum Anda membuat tumpukan Anda.

LaunchTemplateProperti mereferensikan [AWS::EC2::LaunchTemplate](#) sumber daya dengan nama logis myLaunchTemplate yang didefinisikan di tempat lain dalam template Anda.

Note

Untuk contoh templat peluncuran, lihat [Buat template peluncuran dengan CloudFormation](#) di bagian EC2 cuplikan Amazon dan bagian [Contoh](#) di sumber daya. `AWS::EC2::LaunchTemplate`

JSON

```
"myASG" : {
  "Type" : "AWS::AutoScaling::AutoScalingGroup",
  "Properties" : {
    "VPCZoneIdentifier" : [ "subnetIdAz1", "subnetIdAz2", "subnetIdAz3" ],
    "LaunchTemplate" : {
      "LaunchTemplateId" : {
        "Ref" : "myLaunchTemplate"
      },
      "Version" : {
        "Fn::GetAtt" : [
          "myLaunchTemplate",
          "LatestVersionNumber"
        ]
      }
    },
    "MaxSize" : "1",
    "MinSize" : "1"
  }
}
```

YAML

```
myASG:
  Type: AWS::AutoScaling::AutoScalingGroup
  Properties:
    VPCZoneIdentifier:
      - subnetIdAz1
      - subnetIdAz2
```

```

- subnetIdAz3
LaunchTemplate:
  LaunchTemplateId: !Ref myLaunchTemplate
  Version: !GetAtt myLaunchTemplate.LatestVersionNumber
  MaxSize: '1'
  MinSize: '1'

```

Buat grup Auto Scaling dengan penyeimbang beban terlampir

Contoh ini menunjukkan [AWS::AutoScaling::AutoScalingGroup](#) sumber daya untuk load balancing melalui beberapa server. Ini menentukan nama logis AWS sumber daya yang dinyatakan di tempat lain dalam template yang sama.

1. `VPCZoneIdentifier` Properti menentukan nama logis dari dua sumber [AWS::EC2::Subnet](#) daya tempat instance grup EC2 Auto Scaling akan dibuat: dan. `myPublicSubnet1` `myPublicSubnet2`
2. `LaunchTemplate` Properti menentukan [AWS::EC2::LaunchTemplate](#) sumber daya dengan nama `myLaunchTemplate` logis.
3. `TargetGroupARNs` Properti mencantumkan grup target untuk Application Load Balancer atau Network Load Balancer yang digunakan untuk merutekan lalu lintas ke grup Auto Scaling. Dalam contoh ini, satu kelompok target ditentukan, [AWS::ElasticLoadBalancingV2::TargetGroup](#) sumber daya dengan nama logis `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

```

Lihat juga

[Untuk contoh mendetail yang membuat grup Auto Scaling dengan kebijakan penskalaan pelacakan target berdasarkan metrik yang ALBRequestCountPerTarget telah ditentukan untuk Application Load Balancer, lihat bagian Contoh di sumber daya.](#) `AWS::AutoScaling::ScalingPolicy`

Buat grup Auto Scaling dengan notifikasi

Contoh ini menunjukkan [AWS::AutoScaling::AutoScalingGroup](#) sumber daya yang mengirimkan notifikasi Amazon SNS saat peristiwa yang ditentukan berlangsung. `NotificationConfigurations` Properti menentukan topik SNS di mana AWS CloudFormation mengirimkan pemberitahuan dan peristiwa yang akan menyebabkan AWS CloudFormation untuk mengirim pemberitahuan. Ketika peristiwa yang ditentukan oleh `NotificationTypes` terjadi, AWS CloudFormation akan mengirim pemberitahuan ke topik SNS yang ditentukan oleh `TopicARN`. Saat Anda meluncurkan tumpukan, AWS CloudFormation buat [AWS::SNS::Subscription](#) resource (`snsTopicForAutoScalingGroup`) yang dideklarasikan dalam template yang sama.

VPCZoneIdentifierProperti grup Auto Scaling menentukan daftar subnet yang ada di tiga Availability Zone yang berbeda. Anda harus menentukan subnet yang berlaku IDs dari akun Anda sebelum Anda membuat tumpukan Anda. LaunchTemplateProperti mereferensikan nama logis dari [AWS::EC2::LaunchTemplate](#) sumber daya yang dideklarasikan di tempat lain dalam template yang sama.

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

```

Buat grup Auto Scaling yang menggunakan **CreationPolicyUpdatePolicy**

Contoh berikut menunjukkan cara menambahkan CreationPolicy dan UpdatePolicy atribut ke [AWS::AutoScaling::AutoScalingGroup](#) sumber daya.

Kebijakan pembuatan sampel mencegah grup Auto Scaling mencapai CREATE_COMPLETE status hingga AWS CloudFormation menerima Count sejumlah sinyal keberhasilan saat grup siap. Untuk memberi sinyal bahwa grup Auto Scaling sudah siap, skrip cfn-signal helper yang ditambahkan ke data pengguna template peluncuran (tidak ditampilkan) dijalankan pada instance. Jika instance tidak mengirim sinyal dalam yang ditentukanTimeout, CloudFormation mengasumsikan bahwa instance tidak dibuat, pembuatan sumber daya gagal, dan CloudFormation memutar tumpukan kembali.

Kebijakan pembaruan sampel menginstruksikan CloudFormation untuk melakukan pembaruan bergulir menggunakan AutoScalingRollingUpdate properti. Pembaruan bergulir membuat perubahan pada grup Auto Scaling dalam batch kecil (untuk contoh ini, instance demi instance) berdasarkan MaxBatchSize dan waktu jeda antara kumpulan pembaruan berdasarkan PauseTime

`MinInstancesInService` atribut menentukan jumlah minimum instance yang harus dalam layanan dalam grup Auto Scaling CloudFormation sementara memperbarui instance lama.

Atribut `WaitOnResourceSignals` diatur menjadi `true`. CloudFormation harus menerima sinyal dari setiap instans baru dalam `PauseTime` yang ditentukan sebelum melanjutkan pembaruan. Saat pembaruan tumpukan sedang berlangsung, proses EC2 Auto Scaling berikut ditangguhkan: `HealthCheck`, `ReplaceUnhealthy`, `AZRebalanceAlarmNotification`, dan `ScheduledActions`. Catatan: Jangan menangguhkan jenis proses `Launch`, `Terminate`, atau `AddToLoadBalancer` (jika grup Auto Scaling digunakan dengan Elastic Load Balancing) karena hal itu dapat mencegah pembaruan bergulir berfungsi dengan baik.

`VPCZoneIdentifier` properti grup Auto Scaling menentukan daftar subnet yang ada di tiga Availability Zone yang berbeda. Anda harus menentukan subnet yang berlaku IDs dari akun Anda sebelum Anda membuat tumpukan Anda. `LaunchTemplate` properti mereferensikan nama logis dari [AWS::EC2::LaunchTemplate](#) sumber daya yang dideklarasikan di tempat lain dalam template yang sama.

Untuk informasi selengkapnya tentang `UpdatePolicy` atribut `CreationPolicy` dan, lihat [Referensi atribut sumber daya](#).

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'

```

Membuat kebijakan penskalaan langkah

Contoh ini menunjukkan [AWS::AutoScaling::ScalingPolicy](#) sumber daya yang menskalakan grup Auto Scaling menggunakan kebijakan penskalaan langkah. Properti `AdjustmentType` menentukan `ChangeInCapacity`, yang berarti bahwa `ScalingAdjustment` mewakili jumlah instans untuk ditambahkan (jika `ScalingAdjustment` positif) atau dihapus (jika negatif). Dalam contoh ini, `ScalingAdjustment` adalah 1; oleh karena itu, kebijakan menambah jumlah EC2 instance dalam grup sebesar 1 ketika ambang alarm dilanggar.

[AWS::CloudWatch::Alarm](#) Sumber daya `CPUAlarmHigh` menentukan kebijakan penskalaan `ASGScalingPolicyHigh` sebagai tindakan yang akan dijalankan saat alarm berada dalam status ALARM (`AlarmActions`). `Dimensions` Properti mereferensikan nama logis dari [AWS::AutoScaling::AutoScalingGroup](#) sumber daya yang dideklarasikan di tempat lain dalam template yang sama.

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

```

Lihat juga

Untuk contoh templat lainnya untuk kebijakan penskalaan, lihat bagian [Contoh](#) di `AWS::AutoScaling::ScalingPolicy` sumber daya.

Contoh kelompok contoh campuran

Membuat grup Auto Scaling dengan menggunakan pemilihan jenis instans berdasarkan atribut

Contoh ini menunjukkan `AWS::AutoScaling::AutoScalingGroup` sumber daya yang berisi informasi untuk meluncurkan grup instance campuran menggunakan pemilihan tipe instans berbasis atribut. Anda menentukan nilai minimum dan maksimum untuk `VCpuCount` properti dan nilai minimum untuk `MemoryMiB` properti. Tipe instans apa pun yang digunakan oleh grup Auto Scaling harus sesuai dengan atribut instance yang diperlukan.

`VPCZoneIdentifier` properti grup Auto Scaling menentukan daftar subnet yang ada di tiga Availability Zone yang berbeda. Anda harus menentukan subnet yang berlaku IDs dari akun Anda sebelum Anda membuat tumpukan Anda. `LaunchTemplate` properti mereferensikan nama logis dari `AWS::EC2::LaunchTemplate` sumber daya yang dideklarasikan di tempat lain dalam template yang sama.

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'
```

Luncurkan contoh konfigurasi

Buat konfigurasi peluncuran

Contoh ini menunjukkan [AWS::AutoScaling::LaunchConfiguration](#) sumber daya untuk grup Auto Scaling tempat Anda menentukan nilai untuk `ImageId`, `InstanceType`, dan `SecurityGroups` properti. `SecurityGroups` properti menentukan nama logis dari [AWS::EC2::SecurityGroup](#) sumber daya yang ditentukan di tempat lain dalam template, dan grup EC2 keamanan yang ada bernama `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

```

Membuat grup Auto Scaling yang menggunakan konfigurasi peluncuran

Contoh ini menunjukkan [AWS::AutoScaling::AutoScalingGroup](#) sumber daya dengan satu contoh. `VPCZoneIdentifier` properti grup Auto Scaling menentukan daftar subnet yang ada di tiga Availability Zone yang berbeda. Anda harus menentukan subnet yang berlaku IDs dari akun Anda sebelum Anda membuat tumpukan Anda. `LaunchConfigurationName` properti mereferensikan [AWS::AutoScaling::LaunchConfiguration](#) sumber daya dengan nama logis `mySimpleConfig` yang didefinisikan dalam template Anda.

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'
```

Konfigurasi sumber daya Application Auto Scaling dengan AWS CloudFormation

Bagian ini memberikan contoh AWS CloudFormation templat untuk kebijakan penskalaan Application Auto Scaling dan tindakan terjadwal untuk sumber daya yang berbeda. AWS

Important

Ketika cuplikan Application Auto Scaling disertakan dalam template, Anda mungkin perlu mendeklarasikan dependensi pada sumber daya tertentu yang dapat diskalakan yang dibuat melalui template menggunakan atribut. DependsOn ini menimpa paralelisme default dan mengarahkan AWS CloudFormation untuk beroperasi pada sumber daya dalam urutan tertentu. Jika tidak, konfigurasi penskalaan mungkin diterapkan sebelum sumber daya telah disiapkan sepenuhnya.

Untuk informasi lebih lanjut, lihat [Atribut DependsOn](#).

Kategori Snippet

- [Membuat kebijakan penskalaan untuk armada AppStream](#)
- [Buat kebijakan penskalaan untuk kluster Aurora DB](#)
- [Membuat kebijakan penskalaan untuk tabel DynamoDB](#)
- [Membuat kebijakan penskalaan untuk layanan Amazon ECS \(metrik: CPU dan memori rata-rata\)](#)
- [Membuat kebijakan penskalaan untuk layanan Amazon ECS \(metrik: jumlah permintaan rata-rata per target\)](#)
- [Buat tindakan terjadwal dengan ekspresi cron untuk fungsi Lambda](#)
- [Membuat tindakan terjadwal dengan at ekspresi untuk Armada Spot](#)

Membuat kebijakan penskalaan untuk armada AppStream

Cuplikan ini menunjukkan cara membuat kebijakan dan menerapkannya ke [AWS::AppStream::Fleet](#) sumber daya menggunakan sumber daya.

[AWS::ApplicationAutoScaling::ScalingPolicy](#)

[AWS::ApplicationAutoScaling::ScalableTarget](#) Sumber daya mendeklarasikan target yang dapat diskalakan tempat kebijakan ini diterapkan. Application Auto Scaling dapat menskalakan jumlah instance armada minimal 1 instans dan maksimal 20 instans. Kebijakan tersebut menjaga pemanfaatan kapasitas rata-rata armada pada 75 persen, dengan periode cooldown scale-out dan scale-in 300 detik (5 menit).

Ini menggunakan fungsi `Fn::Join` dan `Ref` intrinsik untuk membangun `ResourceId` properti dengan nama logis dari `AWS::AppStream::Fleet` sumber daya yang ditentukan dalam template yang sama. Untuk informasi lebih lanjut, lihat Referensi [fungsi intrinsik](#).

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

```

Buat kebijakan penskalaan untuk kluster Aurora DB

Dalam cuplikan ini, Anda mendaftarkan sumber daya. [AWS::RDS::DBCluster](#)
[AWS::ApplicationAutoScaling::ScalableTarget](#) Sumber daya menunjukkan bahwa
 cluster DB harus diskalakan secara dinamis untuk memiliki dari satu hingga delapan Replika
 Aurora. Anda juga menerapkan kebijakan penskalaan pelacakan target ke kluster menggunakan
[AWS::ApplicationAutoScaling::ScalingPolicy](#) sumber daya.

Dalam konfigurasi ini, metrik standar `RDSReaderAverageCPUUtilization` digunakan untuk
 menyesuaikan kluster DB Aurora berdasarkan pemanfaatan CPU rata-rata sebesar 40 persen di
 seluruh Replika Aurora dalam kluster DB Aurora tersebut. Konfigurasi ini menyediakan periode
 pendinginan penskalaan ke dalam selama 10 menit dan periode pendinginan penskalaan ke luar
 selama 5 menit.

Contoh ini menggunakan fungsi Fn::Sub intrinsik untuk membangun ResourceId properti dengan nama logis AWS::RDS::DBCluster sumber daya yang ditentukan dalam template yang sama. Untuk informasi lebih lanjut, lihat Referensi [fungsi intrinsik](#).

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

```

Membuat kebijakan penskalaan untuk tabel DynamoDB

Cuplikan ini menunjukkan cara membuat kebijakan dengan tipe `TargetTrackingScaling` kebijakan dan menerapkannya ke [AWS::DynamoDB::Table](#) sumber daya menggunakan sumber daya [AWS::ApplicationAutoScaling::ScalingPolicy](#) [AWS::ApplicationAutoScaling::ScalableTarget](#) Sumber daya menyatakan target yang dapat diskalakan yang diterapkan kebijakan ini, dengan minimal lima unit kapasitas tulis dan maksimum 15. Kebijakan penskalaan menskalakan throughput kapasitas tulis tabel untuk mempertahankan pemanfaatan target pada 50 persen berdasarkan metrik `DynamoDBWriteCapacityUtilization` yang sudah ditentukan sebelumnya.

Ini menggunakan fungsi `Fn::Join` dan `Ref` intrinsik untuk membangun `ResourceId` properti dengan nama logis dari `AWS::DynamoDB::Table` sumber daya yang ditentukan dalam template yang sama. Untuk informasi lebih lanjut, lihat Referensi [fungsi intrinsik](#).

Note

Untuk informasi selengkapnya tentang cara membuat AWS CloudFormation template untuk sumber daya DynamoDB, lihat [posting blog Cara menggunakan untuk mengonfigurasi penskalaan otomatis AWS CloudFormation untuk tabel dan indeks Amazon DynamoDB di Blog Database](#). 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

```

Membuat kebijakan penskalaan untuk layanan Amazon ECS (metrik: CPU dan memori rata-rata)

Cuplikan ini menunjukkan cara membuat kebijakan dan menerapkannya ke [AWS::ECS::Service](#) sumber daya menggunakan sumber daya.

[AWS::ApplicationAutoScaling::ScalingPolicy](#)

[AWS::ApplicationAutoScaling::ScalableTarget](#) Sumber daya mendeklarasikan target yang dapat diskalakan tempat kebijakan ini diterapkan. Application Auto Scaling dapat menskalakan jumlah tugas minimal 1 tugas dan maksimal 6.

Hal itu membuat dua kebijakan penskalaan dengan jenis kebijakan `TargetTrackingScaling`. Kebijakan yang digunakan untuk menskalakan layanan ECS berdasarkan penggunaan rata-rata CPU dan memori layanan. Ini menggunakan fungsi `Fn::Join` dan `Ref` intrinsik untuk membangun `ResourceId` properti dengan nama logis dari [AWS::ECS::Cluster](#) (`myContainerCluster`) dan `AWS::ECS::Service` (`myService`) sumber daya yang ditentukan dalam template yang sama. Untuk informasi lebih lanjut, lihat Referensi [fungsi intrinsik](#).

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
```

Membuat kebijakan penskalaan untuk layanan Amazon ECS (metrik: jumlah permintaan rata-rata per target)

Contoh berikut menerapkan kebijakan penskalaan pelacakan target dengan metrik `ALBRequestCountPerTarget` yang telah ditetapkan ke layanan ECS. Kebijakan ini digunakan untuk menambah kapasitas layanan ECS ketika jumlah permintaan per target (per menit) melebihi nilai target. Karena nilai `DisableScaleIn` diatur menjadi `true`, kebijakan pelacakan target tidak akan menghapus kapasitas dari target yang dapat diskalakan.

Ini menggunakan fungsi `Fn::Join` dan `Fn::GetAtt` intrinsik untuk membangun `ResourceLabel` properti dengan nama logis dari

[`AWS::ElasticLoadBalancingV2::LoadBalancer`](#)(`myLoadBalancer`) dan

[`AWS::ElasticLoadBalancingV2::TargetGroup`](#)(`myTargetGroup`) sumber daya yang ditentukan dalam template yang sama. Untuk informasi lebih lanjut, lihat Referensi [fungsi intrinsik](#).

Properti `MaxCapacity` dan `MinCapacity` pada target yang dapat diskalakan dan properti `TargetValue` nilai parameter referensi kebijakan penskalaan yang Anda teruskan ke templat saat membuat atau memperbarui tumpukan.

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
```

Buat tindakan terjadwal dengan ekspresi cron untuk fungsi Lambda

Cuplikan ini mendaftarkan konkurensi yang disediakan untuk fungsi alias () bernama menggunakan sumber daya.

[AWS::Lambda::AliasBLUEAWS::ApplicationAutoScaling::ScalableTarget](#) Hal itu juga membuat tindakan terjadwal dengan jadwal berulang menggunakan ekspresi cron. Zona waktu untuk jadwal berulang adalah UTC.

Ini menggunakan fungsi Ref intrinsik Fn::Join dan intrinsik di RoleARN properti untuk menentukan ARN dari peran terkait layanan. Ini menggunakan fungsi Fn::Sub intrinsik untuk membangun ResourceId properti dengan nama logis dari [AWS::Lambda::Function](#) atau [AWS::Serverless::Function](#) sumber daya yang ditentukan dalam template yang sama. Untuk informasi lebih lanjut, lihat Referensi [fungsi intrinsik](#).

Note

Anda tidak dapat mengalokasikan konkurensi yang disediakan pada alias yang menunjuk ke versi () yang tidak dipublikasikan. \$LATEST

Untuk informasi selengkapnya tentang cara membuat AWS CloudFormation template untuk sumber daya Lambda, lihat posting blog [Menjadwalkan Konkurensi Penyediaan AWS Lambda untuk](#) penggunaan puncak berulang di Blog Komputasi. AWS

JSON

```
{
  "ScalableTarget" : {
    "Type" : "AWS::ApplicationAutoScaling::ScalableTarget",
    "Properties" : {
      "MaxCapacity" : 250,
      "MinCapacity" : 0,
      "RoleARN" : {
        "Fn::Join" : [
          ":",
          [
            "arn:aws:iam:",
            {
```



```

        "Ref" : "AWS::AccountId"
      },
      "role/aws-service-role/lambda.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_LambdaConcurrency"
    ]
  ]
},
"ServiceNamespace" : "lambda",
"ScalableDimension" : "lambda:function:ProvisionedConcurrency",
"ResourceId" : { "Fn::Sub" : "function:${logicalName}:BLUE" },
"ScheduledActions" : [
  {
    "ScalableTargetAction" : {
      "MinCapacity" : "250"
    },
    "ScheduledActionName" : "my-scale-out-scheduled-action",
    "Schedule" : "cron(0 18 * * ? *)",
    "EndTime" : "2022-12-31T12:00:00.000Z"
  }
]
}
}
}
}

```

YAML

```

ScalableTarget:
  Type: AWS::ApplicationAutoScaling::ScalableTarget
  Properties:
    MaxCapacity: 250
    MinCapacity: 0
    RoleARN: !Join
      - ':'
      - - 'arn:aws:iam:'
        - !Ref 'AWS::AccountId'
        - role/aws-service-role/lambda.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_LambdaConcurrency
    ServiceNamespace: lambda
    ScalableDimension: lambda:function:ProvisionedConcurrency
    ResourceId: !Sub function:${logicalName}:BLUE
    ScheduledActions:
      - ScalableTargetAction:
          MinCapacity: 250

```

```
ScheduledActionName: my-scale-out-scheduled-action
Schedule: 'cron(0 18 * * ? *)'
EndTime: '2022-12-31T12:00:00.000Z'
```

Membuat tindakan terjadwal dengan **at** ekspresi untuk Armada Spot

Cuplikan ini menunjukkan cara membuat dua tindakan terjadwal yang terjadi hanya sekali untuk [AWS::EC2::SpotFleet](#) sumber daya yang menggunakan sumber daya. [AWS::ApplicationAutoScaling::ScalableTarget](#) Zona waktu untuk setiap tindakan terjadwal satu kali adalah UTC.

Ini menggunakan fungsi `Fn::Join` dan `Ref` intrinsik untuk membangun `ResourceId` properti dengan nama logis dari `AWS::EC2::SpotFleet` sumber daya yang ditentukan dalam template yang sama. Untuk informasi lebih lanjut, lihat Referensi [fungsi intrinsik](#).

Note

Permintaan Armada Spot harus memiliki jenis permintaan `maintain`. Penskalaan otomatis tidak didukung untuk permintaan tipe `one-time`, atau blok Spot.

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 Cuplikan templat Konsol Penagihan

Contoh ini membuat satu paket harga dengan aturan harga markup global 10%. Paket harga ini dilampirkan ke grup penagihan. Grup penagihan juga memiliki dua item baris khusus yang menerapkan biaya \$10 dan biaya 10% di atas total biaya grup penagihan.

JSON

```

{
  "Parameters": {
    "LinkedAccountIds": {
      "Type": "ListNumber"
    },
    "PrimaryAccountId": {
      "Type": "Number"
    }
  },
  "Resources": {
    "TestPricingRule": {
      "Type": "AWS::BillingConductor::PricingRule",
      "Properties": {
        "Name": "TestPricingRule",
        "Description": "Test pricing rule created through Cloudformation. Mark
everything by 10%.",
        "Type": "MARKUP",
        "Scope": "GLOBAL",
        "ModifierPercentage": 10
      }
    }
  }
}

```

```

    },
    "TestPricingPlan": {
      "Type": "AWS::BillingConductor::PricingPlan",
      "Properties": {
        "Name": "TestPricingPlan",
        "Description": "Test pricing plan created through Cloudformation.",
        "PricingRuleArns": [
          {"Fn::GetAtt": ["TestPricingRule", "Arn"]}
        ]
      }
    },
    "TestBillingGroup": {
      "Type": "AWS::BillingConductor::BillingGroup",
      "Properties": {
        "Name": "TestBillingGroup",
        "Description": "Test billing group created through Cloudformation with 1
linked account. The linked account is also the primary account.",
        "PrimaryAccountId": {
          "Ref": "PrimaryAccountId"
        },
        "AccountGrouping": {
          "LinkedAccountIds": null
        },
        "ComputationPreference": {
          "PricingPlanArn": {
            "Fn::GetAtt": ["TestPricingPlan", "Arn"]
          }
        }
      }
    },
    "TestFlatCustomLineItem": {
      "Type": "AWS::BillingConductor::CustomLineItem",
      "Properties": {
        "Name": "TestFlatCustomLineItem",
        "Description": "Test flat custom line item created through Cloudformation
for a $10 charge.",
        "BillingGroupArn": {
          "Fn::GetAtt": ["TestBillingGroup", "Arn"]
        },
        "CustomLineItemChargeDetails": {
          "Flat": {
            "ChargeValue": 10
          },
          "Type": "FEE"
        }
      }
    }
  }
}

```

```

    }
  },
  "TestPercentageCustomLineItem": {
    "Type": "AWS::BillingConductor::CustomLineItem",
    "Properties": {
      "Name": "TestPercentageCustomLineItem",
      "Description": "Test percentage custom line item created through
Cloudformation for a %10 additional charge on the overall total bill of the billing
group.",
      "BillingGroupArn": {
        "Fn::GetAtt": ["TestBillingGroup", "Arn"]
      },
      "CustomLineItemChargeDetails": {
        "Percentage": {
          "PercentageValue": 10,
          "ChildAssociatedResources": [
            {"Fn::GetAtt": ["TestBillingGroup", "Arn"]}
          ]
        },
        "Type": "FEE"
      }
    }
  }
}

```

YAML

```

Parameters:
  LinkedAccountIds:
    Type: ListNumber
  PrimaryAccountId:
    Type: Number
Resources:
  TestPricingRule:
    Type: 'AWS::BillingConductor::PricingRule'
    Properties:
      Name: 'TestPricingRule'
      Description: 'Test pricing rule created through Cloudformation. Mark everything
by 10%.'
      Type: 'MARKUP'
      Scope: 'GLOBAL'

```

```
    ModifierPercentage: 10
TestPricingPlan:
  Type: 'AWS::BillingConductor::PricingPlan'
  Properties:
    Name: 'TestPricingPlan'
    Description: 'Test pricing plan created through Cloudformation.'
    PricingRuleArns:
      - !GetAtt TestPricingRule.Arn
TestBillingGroup:
  Type: 'AWS::BillingConductor::BillingGroup'
  Properties:
    Name: 'TestBillingGroup'
    Description: 'Test billing group created through Cloudformation with 1 linked
account. The linked account is also the primary account.'
    PrimaryAccountId: !Ref PrimaryAccountId
    AccountGrouping:
      LinkedAccountIds: !Ref LinkedAccountIds
    ComputationPreference:
      PricingPlanArn: !GetAtt TestPricingPlan.Arn
TestFlatCustomLineItem:
  Type: 'AWS::BillingConductor::CustomLineItem'
  Properties:
    Name: 'TestFlatCustomLineItem'
    Description: 'Test flat custom line item created through Cloudformation for a $10
charge.'
    BillingGroupArn: !GetAtt TestBillingGroup.Arn
    CustomLineItemChargeDetails:
      Flat:
        ChargeValue: 10
        Type: 'FEE'
TestPercentageCustomLineItem:
  Type: 'AWS::BillingConductor::CustomLineItem'
  Properties:
    Name: 'TestPercentageCustomLineItem'
    Description: 'Test percentage custom line item created through Cloudformation for
a %10 additional charge on the overall total bill of the billing group.'
    BillingGroupArn: !GetAtt TestBillingGroup.Arn
    CustomLineItemChargeDetails:
      Percentage:
        PercentageValue: 10
      ChildAssociatedResources:
        - !GetAtt TestBillingGroup.Arn
    Type: 'FEE'
```

AWS CloudFormation cuplikan template

Topik

- [Tumpukan nested](#)
- [Kondisi menunggu](#)

Tumpukan nested

Membuat nest tumpukan dalam templat

Contoh templat ini mengandung sumber daya tumpukan nested yang disebut myStack. Ketika AWS CloudFormation membuat tumpukan dari template, itu menciptakan myStack, yang templatnya ditentukan dalam TemplateURL properti. Nilai output StackRef mengembalikan ID tumpukan untuk myStack dan nilai OutputFromNestedStack mengembalikan nilai output BucketName dari dalam sumber daya myStack. Outputs.*nestedstackoutputname*Format ini dicadangkan untuk menentukan nilai output dari tumpukan bersarang dan dapat digunakan di mana saja dalam template yang berisi.

Untuk informasi selengkapnya, lihat [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
```

Membuat nest tumpukan dengan parameter input dalam sebuah templat

Templat contoh ini berisi sumber daya tumpukan yang menentukan parameter input. Saat AWS CloudFormation membuat tumpukan dari template ini, ia menggunakan pasangan nilai yang dideklarasikan dalam Parameters properti sebagai parameter input untuk template yang digunakan untuk membuat myStackWithParams tumpukan. Dalam contoh ini, parameter InstanceType dan KeyName ditentukan.

Untuk informasi selengkapnya, lihat [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

```

Kondisi menunggu

Menggunakan kondisi tunggu dengan EC2 instans Amazon

Important

Untuk sumber daya Amazon EC2 dan Auto Scaling, sebaiknya gunakan `CreationPolicy` atribut alih-alih kondisi tunggu. Tambahkan `CreationPolicy` atribut ke sumber daya tersebut, dan gunakan skrip pembantu sinyal cfn untuk memberi sinyal ketika proses pembuatan instance telah berhasil diselesaikan.

Jika Anda tidak dapat menggunakan kebijakan pembuatan, Anda akan melihat contoh template berikut, yang mendeklarasikan EC2 instance Amazon dengan kondisi tunggu. Kondisi `myWaitCondition` tunggu digunakan `myWaitConditionHandle` untuk pensinyalan, menggunakan `DependsOn` atribut untuk menentukan bahwa kondisi tunggu akan dipicu setelah sumber daya EC2 instans Amazon dibuat, dan menggunakan `Timeout` properti untuk menentukan durasi 4500 detik untuk kondisi tunggu. Selain itu, URL yang telah ditetapkan sebelumnya yang memberi sinyal kondisi tunggu diteruskan ke EC2 instance Amazon dengan `UserData` properti `Ec2Instance` sumber daya, sehingga memungkinkan aplikasi atau skrip yang berjalan pada EC2 instance Amazon tersebut untuk mengambil URL yang telah ditetapkan sebelumnya dan menggunakannya untuk memberi sinyal keberhasilan atau kegagalan kondisi tunggu. Anda perlu

menggunakan `cfn-signal` atau membuat aplikasi atau skrip yang menandakan kondisi tunggu. Nilai output `ApplicationData` berisi data yang diteruskan kembali dari sinyal kondisi tunggu.

Untuk informasi selengkapnya, lihat [Buat kondisi tunggu di CloudFormation template](#).

JSON

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Mappings" : {
    "RegionMap" : {
      "us-east-1" : {
        "AMI" : "ami-0ff8a91507f77f867"
      },
      "us-west-1" : {
        "AMI" : "ami-0bdb828fd58c52235"
      },
      "eu-west-1" : {
        "AMI" : "ami-047bb4163c506cd98"
      },
      "ap-northeast-1" : {
        "AMI" : "ami-06cd52961ce9f0d85"
      },
      "ap-southeast-1" : {
        "AMI" : "ami-08569b978cc4dfa10"
      }
    }
  },
  "Resources" : {
    "Ec2Instance" : {
      "Type" : "AWS::EC2::Instance",
      "Properties" : {
        "UserData" : { "Fn::Base64" : {"Ref" : "myWaitHandle"} },
        "ImageId" : { "Fn::FindInMap" : [ "RegionMap", { "Ref" :
"AWS::Region" }, "AMI" ] }
      }
    },
    "myWaitHandle" : {
      "Type" : "AWS::CloudFormation::WaitConditionHandle",
      "Properties" : {
      }
    },
    "myWaitCondition" : {
      "Type" : "AWS::CloudFormation::WaitCondition",
```

```

        "DependsOn" : "Ec2Instance",
        "Properties" : {
            "Handle" : { "Ref" : "myWaitHandle" },
            "Timeout" : "4500"
        }
    },
    "Outputs" : {
        "ApplicationData" : {
            "Value" : { "Fn::GetAtt" : [ "myWaitCondition", "Data" ] },
            "Description" : "The data passed back as part of signalling the
WaitCondition."
        }
    }
}

```

YAML

```

AWSTemplateFormatVersion: '2010-09-09'
Mappings:
  RegionMap:
    us-east-1:
      AMI: ami-0ff8a91507f77f867
    us-west-1:
      AMI: ami-0bdb828fd58c52235
    eu-west-1:
      AMI: ami-047bb4163c506cd98
    ap-northeast-1:
      AMI: ami-06cd52961ce9f0d85
    ap-southeast-1:
      AMI: ami-08569b978cc4dfa10
Resources:
  Ec2Instance:
    Type: AWS::EC2::Instance
    Properties:
      UserData:
        Fn::Base64: !Ref myWaitHandle
      ImageId:
        Fn::FindInMap:
          - RegionMap
          - Ref: AWS::Region
          - AMI
  myWaitHandle:

```

```

    Type: AWS::CloudFormation::WaitConditionHandle
    Properties: {}
  myWaitCondition:
    Type: AWS::CloudFormation::WaitCondition
    DependsOn: Ec2Instance
    Properties:
      Handle: !Ref myWaitHandle
      Timeout: '4500'
  Outputs:
    ApplicationData:
      Value: !GetAtt myWaitCondition.Data
      Description: The data passed back as part of signalling the WaitCondition.

```

Menggunakan skrip pembantu cfn-signal untuk memberi sinyal kondisi menunggu

Contoh ini menunjukkan baris `cfn-signal` perintah yang menandakan keberhasilan untuk kondisi menunggu. Anda perlu menentukan baris perintah di `UserData` properti EC2 instance.

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
    - |
    ;
```

Menggunakan Curl untuk memberikan sinyal kondisi menunggu

Contoh ini menunjukkan baris perintah Curl yang memberikan sinyal berhasil ke kondisi menunggu.

```
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"
```

Tempat file /tmp/a mengandung struktur JSON berikut:

```
{
  "Status" : "SUCCESS",
  "Reason" : "Configuration Complete",
  "UniqueId" : "ID1234",
  "Data" : "Application has completed configuration."
}
```

Contoh ini menunjukkan baris perintah Curl yang mengirimkan sinyal berhasil yang sama tetapi mengirimkan JSON sebagai parameter pada baris perintah.

```
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"
```

Cuplikan CloudFront template Amazon

Gunakan contoh cuplikan templat ini dengan sumber daya CloudFront distribusi Amazon Anda di AWS CloudFormation Untuk informasi selengkapnya, lihat [referensi jenis CloudFront sumber daya Amazon](#).

Topik

- [Sumber daya CloudFront distribusi Amazon dengan asal Amazon S3](#)
- [Sumber daya CloudFront distribusi Amazon dengan asal kustom](#)
- [CloudFront Distribusi Amazon dengan dukungan multi-asal](#)
- [CloudFront Distribusi Amazon dengan fungsi Lambda sebagai asal](#)
- [Lihat juga](#)

Sumber daya CloudFront distribusi Amazon dengan asal Amazon S3

Contoh template berikut menunjukkan CloudFront [Distribusi](#) Amazon menggunakan [S3Origin dan identitas akses asal](#) lama (OAI). Untuk informasi tentang menggunakan kontrol akses asal (OAC) sebagai gantinya, lihat [Membatasi akses ke asal Layanan Penyimpanan Sederhana Amazon di Panduan CloudFront](#) Pengembang Amazon.

JSON

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Resources" : {
    "myDistribution" : {
      "Type" : "AWS::CloudFront::Distribution",
      "Properties" : {
        "DistributionConfig" : {
          "Origins" : [ {
            "DomainName" : "amzn-s3-demo-bucket.s3.amazonaws.com",
            "Id" : "myS3Origin",
            "S3OriginConfig" : {
              "OriginAccessIdentity" : "origin-access-identity/
cloudfront/E127EXAMPLE51Z"
            }
          }
        ],
        "Enabled" : "true",
        "Comment" : "Some comment",
```

```
"DefaultRootObject" : "index.html",
"Logging" : {
    "IncludeCookies" : "false",
    "Bucket" : "amzn-s3-demo-logging-bucket.s3.amazonaws.com",
    "Prefix" : "myprefix"
},
"Aliases" : [ "mysite.example.com", "yoursite.example.com" ],
"DefaultCacheBehavior" : {
    "AllowedMethods" : [ "DELETE", "GET", "HEAD", "OPTIONS",
"PATCH", "POST", "PUT" ],
    "TargetOriginId" : "myS3Origin",
    "ForwardedValues" : {
        "QueryString" : "false",
        "Cookies" : { "Forward" : "none" }
    },
    "TrustedSigners" : [ "1234567890EX", "1234567891EX" ],
    "ViewerProtocolPolicy" : "allow-all"
},
"PriceClass" : "PriceClass_200",
"Restrictions" : {
    "GeoRestriction" : {
        "RestrictionType" : "whitelist",
        "Locations" : [ "AQ", "CV" ]
    }
},
"ViewerCertificate" : { "CloudFrontDefaultCertificate" : "true" }
}
}
}
```

YAML

```
AWSTemplateFormatVersion: '2010-09-09'
Resources:
  myDistribution:
    Type: AWS::CloudFront::Distribution
    Properties:
      DistributionConfig:
        Origins:
          - DomainName: amzn-s3-demo-bucket.s3.amazonaws.com
            Id: myS3Origin
```



```
S3OriginConfig:
  OriginAccessIdentity: origin-access-identity/cloudfront/E127EXAMPLE51Z
Enabled: 'true'
Comment: Some comment
DefaultRootObject: index.html
Logging:
  IncludeCookies: 'false'
  Bucket: amzn-s3-demo-logging-bucket.s3.amazonaws.com
  Prefix: myprefix
Aliases:
- mysite.example.com
- yoursite.example.com
DefaultCacheBehavior:
  AllowedMethods:
  - DELETE
  - GET
  - HEAD
  - OPTIONS
  - PATCH
  - POST
  - PUT
  TargetOriginId: myS3Origin
  ForwardedValues:
    QueryString: 'false'
    Cookies:
      Forward: none
  TrustedSigners:
  - 1234567890EX
  - 1234567891EX
  ViewerProtocolPolicy: allow-all
PriceClass: PriceClass_200
Restrictions:
  GeoRestriction:
    RestrictionType: whitelist
    Locations:
    - AQ
    - CV
ViewerCertificate:
  CloudFrontDefaultCertificate: 'true'
```

Sumber daya CloudFront distribusi Amazon dengan asal kustom

Contoh template berikut menunjukkan CloudFront [Distribusi](#) Amazon menggunakan file [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

```

AWS::CloudFront::Distribution
  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 Distribusi Amazon dengan dukungan multi-asal

Contoh template berikut menunjukkan cara mendeklarasikan CloudFront [Distribusi dengan dukungan multi-asal](#). Dalam [DistributionConfig](#), daftar asal disediakan dan a [DefaultCacheBehavior](#) ditetapkan.

JSON

```

{
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Resources" : {

```

```

    "myDistribution" : {
      "Type" : "AWS::CloudFront::Distribution",
      "Properties" : {
        "DistributionConfig" : {
          "Origins" : [ {
            "Id" : "myS3Origin",
            "DomainName" : "amzn-s3-demo-bucket.s3.amazonaws.com",
            "S3OriginConfig" : {
              "OriginAccessIdentity" : "origin-access-identity/
cloudfront/E127EXAMPLE51Z"
            }
          },
          {
            "Id" : "myCustomOrigin",
            "DomainName" : "www.example.com",
            "CustomOriginConfig" : {
              "HTTPPort" : "80",
              "HTTPSPort" : "443",
              "OriginProtocolPolicy" : "http-only"
            }
          }
        ],
        "Enabled" : "true",
        "Comment" : "Some comment",
        "DefaultRootObject" : "index.html",
        "Logging" : {
          "IncludeCookies" : "true",
          "Bucket" : "amzn-s3-demo-logging-bucket.s3.amazonaws.com",
          "Prefix" : "myprefix"
        },
        "Aliases" : [ "mysite.example.com", "yoursite.example.com" ],
        "DefaultCacheBehavior" : {
          "TargetOriginId" : "myS3Origin",
          "ForwardedValues" : {
            "QueryString" : "false",
            "Cookies" : { "Forward" : "all" }
          },
          "TrustedSigners" : [ "1234567890EX", "1234567891EX" ],
          "ViewerProtocolPolicy" : "allow-all",
          "MinTTL" : "100",
          "SmoothStreaming" : "true"
        },
        "CacheBehaviors" : [ {

```

```

        "AllowedMethods" : [ "DELETE", "GET", "HEAD", "OPTIONS",
"PATCH", "POST", "PUT" ],
        "TargetOriginId" : "myS3Origin",
        "ForwardedValues" : {
            "QueryString" : "true",
            "Cookies" : { "Forward" : "none" }
        },
        "TrustedSigners" : [ "1234567890EX", "1234567891EX" ],
        "ViewerProtocolPolicy" : "allow-all",
        "MinTTL" : "50",
        "PathPattern" : "images1/*.jpg"
    },
    {
        "AllowedMethods" : [ "DELETE", "GET", "HEAD", "OPTIONS",
"PATCH", "POST", "PUT" ],
        "TargetOriginId" : "myCustomOrigin",
        "ForwardedValues" : {
            "QueryString" : "true",
            "Cookies" : { "Forward" : "none" }
        },
        "TrustedSigners" : [ "1234567890EX", "1234567891EX" ],
        "ViewerProtocolPolicy" : "allow-all",
        "MinTTL" : "50",
        "PathPattern" : "images2/*.jpg"
    }
],
"CustomErrorResponses" : [ {
    "ErrorCode" : "404",
    "ResponsePagePath" : "/error-pages/404.html",
    "ResponseCode" : "200",
    "ErrorCachingMinTTL" : "30"
} ],
"PriceClass" : "PriceClass_All",
"ViewerCertificate" : { "CloudFrontDefaultCertificate" : "true" }
}
}
}
}
}
}
}
}
}
}

```

YAML

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

Resources:**myDistribution:****Type:** AWS::CloudFront::Distribution**Properties:****DistributionConfig:****Origins:**

- Id: myS3Origin

DomainName: amzn-s3-demo-bucket.s3.amazonaws.com**S3OriginConfig:****OriginAccessIdentity:** origin-access-identity/cloudfront/E127EXAMPLE51Z

- Id: myCustomOrigin

DomainName: www.example.com**CustomOriginConfig:****HTTPPort:** '80'**HTTPSPort:** '443'**OriginProtocolPolicy:** http-only**Enabled:** 'true'**Comment:** Some comment**DefaultRootObject:** index.html**Logging:****IncludeCookies:** 'true'**Bucket:** amzn-s3-demo-logging-bucket.s3.amazonaws.com**Prefix:** myprefix**Aliases:**

- mysite.example.com
- yoursite.example.com

DefaultCacheBehavior:**TargetOriginId:** myS3Origin**ForwardedValues:****QueryString:** 'false'**Cookies:****Forward:** all**TrustedSigners:**

- 1234567890EX
- 1234567891EX

ViewerProtocolPolicy: allow-all**MinTTL:** '100'**SmoothStreaming:** 'true'**CacheBehaviors:**

- AllowedMethods:

- DELETE
- GET
- HEAD
- OPTIONS

```
- PATCH
- POST
- PUT
TargetOriginId: myS3Origin
ForwardedValues:
  QueryString: 'true'
  Cookies:
    Forward: none
TrustedSigners:
- 1234567890EX
- 1234567891EX
ViewerProtocolPolicy: allow-all
MinTTL: '50'
PathPattern: images1/*.jpg
- AllowedMethods:
- DELETE
- GET
- HEAD
- OPTIONS
- PATCH
- POST
- PUT
TargetOriginId: myCustomOrigin
ForwardedValues:
  QueryString: 'true'
  Cookies:
    Forward: none
TrustedSigners:
- 1234567890EX
- 1234567891EX
ViewerProtocolPolicy: allow-all
MinTTL: '50'
PathPattern: images2/*.jpg
CustomErrorResponses:
- ErrorCode: '404'
  ResponsePagePath: "/error-pages/404.html"
  ResponseCode: '200'
  ErrorCachingMinTTL: '30'
PriceClass: PriceClass_All
ViewerCertificate:
  CloudFrontDefaultCertificate: 'true'
```


CloudFront Distribusi Amazon dengan fungsi Lambda sebagai asal

Contoh berikut membuat CloudFront distribusi yang menampilkan URL fungsi Lambda tertentu (disediakan sebagai parameter), memungkinkan akses khusus HTTP, caching, kompresi, dan pengiriman global. Ini mengonfigurasi URL Lambda sebagai asal HTTPS khusus dan menerapkan kebijakan caching AWS standar. Distribusi dioptimalkan untuk kinerja dengan HTTP/2 dan IPv6 mendukung dan mengeluarkan nama CloudFront domain, memungkinkan pengguna untuk mengakses fungsi Lambda melalui titik akhir yang aman dan didukung CDN-nya. Untuk informasi selengkapnya, lihat [Menggunakan Amazon CloudFront dengan AWS Lambda sebagai asal untuk mempercepat aplikasi web Anda](#) di AWS Blog.

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}/
```

Lihat juga

Untuk contoh menambahkan alias kustom ke catatan Route 53 untuk membuat nama ramah untuk CloudFront distribusi, lihat [Catatan sumber daya alias ditetapkan untuk distribusi CloudFront](#).

Cuplikan CloudWatch template Amazon

Gunakan contoh cuplikan templat ini untuk membantu mendeskripsikan CloudWatch sumber daya Amazon Anda. AWS CloudFormation Untuk informasi selengkapnya, lihat [referensi jenis CloudWatch sumber daya Amazon](#).

Topik

- [Alarm penagihan](#)
- [Alarm penggunaan CPU](#)
- [Memulihkan instans Amazon Elastic Compute Cloud](#)
- [Membuat dasbor dasar](#)
- [Buat dasbor dengan side-by-side widget](#)

Alarm penagihan

Dalam contoh berikut, Amazon CloudWatch mengirimkan pemberitahuan email ketika tagihan ke AWS akun Anda melebihi ambang batas alarm. Untuk menerima notifikasi penggunaan, aktifkan peringatan penagihan. Untuk informasi selengkapnya, lihat [Membuat alarm penagihan untuk memantau perkiraan AWS tagihan Anda](#) di Panduan CloudWatch Pengguna Amazon. >

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"

```

Alarm penggunaan CPU

Contoh cuplikan berikut membuat alarm yang mengirimkan notifikasi ketika penggunaan CPU rata-rata dari EC2 instans Amazon melebihi 90 persen selama lebih dari 60 detik selama tiga periode evaluasi.

JSON

```

"CPUALarm" : {
  "Type" : "AWS::CloudWatch::Alarm",
  "Properties" : {
    "AlarmDescription" : "CPU alarm for my instance",
    "AlarmActions" : [ { "Ref" : "logical name of an AWS::SNS::Topic resource" } ],
    "MetricName" : "CPUUtilization",
    "Namespace" : "AWS/EC2",
    "Statistic" : "Average",
    "Period" : "60",
    "EvaluationPeriods" : "3",
    "Threshold" : "90",
    "ComparisonOperator" : "GreaterThanThreshold",
    "Dimensions" : [ {
      "Name" : "InstanceId",
      "Value" : { "Ref" : "logical name of an AWS::EC2::Instance resource" }
    } ]
  }
}

```

YAML

```

CPUALarm:
  Type: AWS::CloudWatch::Alarm
  Properties:
    AlarmDescription: CPU alarm for my instance

```

```

AlarmActions:
- !Ref: "logical name of an AWS::SNS::Topic resource"
MetricName: CPUUtilization
Namespace: AWS/EC2
Statistic: Average
Period: '60'
EvaluationPeriods: '3'
Threshold: '90'
ComparisonOperator: GreaterThanThreshold
Dimensions:
- Name: InstanceId
  Value: !Ref: "logical name of an AWS::EC2::Instance resource"

```

Memulihkan instans Amazon Elastic Compute Cloud

CloudWatch Alarm berikut memulihkan EC2 instance ketika mengalami kegagalan pemeriksaan status selama 15 menit berturut-turut. Untuk informasi selengkapnya tentang tindakan alarm, lihat [Membuat alarm untuk menghentikan, menghentikan, mem-boot ulang, atau memulihkan EC2 instance](#) di CloudWatch Panduan Pengguna Amazon.

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

```

Membuat dasbor dasar

Contoh berikut membuat CloudWatch dasbor sederhana dengan satu widget metrik yang menampilkan pemanfaatan CPU, dan satu widget teks menampilkan pesan.

JSON

```

{
  "BasicDashboard": {
    "Type": "AWS::CloudWatch::Dashboard",

```

```

    "Properties": {
      "DashboardName": "Dashboard1",
      "DashboardBody": "{\"widgets\": [{\"type\": \"metric\", \"x\": 0, \"y\": 0, \"width\": 12, \"height\": 6, \"properties\": {\"metrics\": [\"AWS/EC2\", \"CPUUtilization\", \"InstanceId\", \"i-012345\"], \"period\": 300, \"stat\": \"Average\", \"region\": \"us-east-1\", \"title\": \"EC2 Instance CPU\"}}, {\"type\": \"text\", \"x\": 0, \"y\": 7, \"width\": 3, \"height\": 3, \"properties\": {\"markdown\": \"Hello world\"}}] }"
    }
  }
}

```

YAML

```

BasicDashboard:
  Type: AWS::CloudWatch::Dashboard
  Properties:
    DashboardName: Dashboard1
    DashboardBody: '{"widgets":
[{"type":"metric","x":0,"y":0,"width":12,"height":6,"properties":{"metrics":["AWS/
EC2","CPUUtilization","InstanceId","i-012345"],"period":300,"stat":"Average","region":"us-
east-1","title":"EC2 Instance CPU"}},
{"type":"text","x":0,"y":7,"width":3,"height":3,"properties":{"markdown":"Hello
world"}}]}'

```

Buat dasbor dengan side-by-side widget

Contoh berikut membuat dasbor dengan dua widget metrik yang muncul berdampingan.

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"}]}]}'
```

Cuplikan templat Amazon CloudWatch Log

Amazon CloudWatch Logs dapat memantau sistem, aplikasi, dan file log kustom Anda dari EC2 instans Amazon atau sumber lain. Anda dapat menggunakan AWS CloudFormation untuk menyediakan dan mengelola grup log dan filter metrik. Untuk informasi selengkapnya tentang CloudWatch Log Amazon, lihat [Panduan Pengguna CloudWatch Log Amazon](#).

Topik

- [Kirim CloudWatch log ke Log dari instance Linux](#)
- [Kirim CloudWatch log ke Log dari instance Windows](#)
- [Lihat juga](#)

Kirim CloudWatch log ke Log dari instance Linux

Templat berikut menjelaskan server web dan metrik kustomnya. Log acara dari log server web menyediakan data untuk metrik kustom. Untuk mengirim peristiwa log ke metrik kustom, UserData bidang menginstal agen CloudWatch Log di EC2 instans Amazon. Informasi konfigurasi untuk agen, seperti lokasi berkas log server, nama grup log, dan nama aliran log, ditentukan di file /tmp/

`cwlogs/apacheaccess.conf`. Aliran log dibuat setelah server web mulai mengirim log acara ke file `/var/log/httpd/access_log`.

Note

Catatan tentang izin: Instans `WebServerHost` mereferensikan profil instans `LogRoleInstanceProfile`, yang pada gilirannya mereferensikan peran `LogRole`. `LogRole` menentukan izin `s3:GetObject` untuk `arn:aws:s3:::`. Izin ini diperlukan karena `WebServerHost` mengunduh agen `CloudWatch Log` (`awslogs-agent-setup.py`) dari Amazon S3 di bagian. `UserData`

Dua filter metrik menjelaskan cara informasi log diubah menjadi metrik `CloudWatch`. Metrik `404` menghitung jumlah kejadian `404`. Metrik ukuran melacak ukuran permintaan. Kedua `CloudWatch` alarm akan mengirimkan pemberitahuan jika ada lebih dari dua `404` dalam waktu 2 menit atau jika ukuran permintaan rata-rata lebih dari `3500 KB` selama 10 menit.

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "AWS CloudFormation Sample Template for CloudWatch Logs.",
  "Parameters": {
    "KeyName": {
      "Description": "Name of an existing EC2 KeyPair to enable SSH access to the instances",
      "Type": "AWS::EC2::KeyPair::KeyName",
      "ConstraintDescription": "must be the name of an existing EC2 KeyPair."
    },
    "SSHLocation": {
      "Description": "The IP address range that can be used to SSH to the EC2 instances",
      "Type": "String",
      "MinLength": "9",
      "MaxLength": "18",
      "Default": "0.0.0.0/0",
      "AllowedPattern": "(\\d{1,3})\\. (\\d{1,3})\\. (\\d{1,3})\\. (\\d{1,3})/(\\d{1,2})",
      "ConstraintDescription": "must be a valid IP CIDR range of the form x.x.x.x/x."
    }
  },
```

```

    "OperatorEmail": {
      "Description": "Email address to notify if there are any scaling
operations",
      "Type": "String"
    }
  },
  "Mappings": {
    "RegionMap": {
      "us-east-1": {
        "AMI": "ami-0ff8a91507f77f867"
      },
      "us-west-1": {
        "AMI": "ami-0bdb828fd58c52235"
      },
      "us-west-2": {
        "AMI": "ami-a0cfeed8"
      },
      "eu-west-1": {
        "AMI": "ami-047bb4163c506cd98"
      },
      "ap-southeast-1": {
        "AMI": "ami-08569b978cc4dfa10"
      },
      "ap-southeast-2": {
        "AMI": "ami-09b42976632b27e9b"
      },
      "ap-northeast-1": {
        "AMI": "ami-06cd52961ce9f0d85"
      },
      "sa-east-1": {
        "AMI": "ami-07b14488da8ea02a0"
      },
      "eu-central-1": {
        "AMI": "ami-0233214e13e500f77"
      }
    }
  },
  "Resources": {
    "LogRole": {
      "Type": "AWS::IAM::Role",
      "Properties": {
        "AssumeRolePolicyDocument": {
          "Version": "2012-10-17",
          "Statement": [

```

```

        {
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "ec2.amazonaws.com"
                ]
            },
            "Action": [
                "sts:AssumeRole"
            ]
        }
    ],
    "Path": "/",
    "Policies": [
        {
            "PolicyName": "LogRolePolicy",
            "PolicyDocument": {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Effect": "Allow",
                        "Action": [
                            "logs:Create*",
                            "logs:PutLogEvents",
                            "s3:GetObject"
                        ],
                        "Resource": [
                            "arn:aws:logs:*:*:*",
                            "arn:aws:s3:::*"
                        ]
                    }
                ]
            }
        }
    ]
},
"LogRoleInstanceProfile": {
    "Type": "AWS::IAM::InstanceProfile",
    "Properties": {
        "Path": "/",
        "Roles": [
            {

```

```

        "Ref": "LogRole"
      }
    ]
  },
  "WebServerSecurityGroup": {
    "Type": "AWS::EC2::SecurityGroup",
    "Properties": {
      "GroupDescription": "Enable HTTP access via port 80 and SSH access via
port 22",
      "SecurityGroupIngress": [
        {
          "IpProtocol": "tcp",
          "FromPort": 80,
          "ToPort": 80,
          "CidrIp": "0.0.0.0/0"
        },
        {
          "IpProtocol": "tcp",
          "FromPort": 22,
          "ToPort": 22,
          "CidrIp": {
            "Ref": "SSHLocation"
          }
        }
      ]
    }
  },
  "WebServerHost": {
    "Type": "AWS::EC2::Instance",
    "Metadata": {
      "Comment": "Install a simple PHP application",
      "AWS::CloudFormation::Init": {
        "config": {
          "packages": {
            "yum": {
              "httpd": [],
              "php": []
            }
          },
          "files": {
            "/tmp/cwlogs/apacheaccess.conf": {
              "content": {
                "Fn::Join": [

```

```

        ""
        [
            "[general]\n",
            "state_file= /var/awslogs/agent-state\n",
            "[/var/log/httpd/access_log]\n",
            "file = /var/log/httpd/access_log\n",
            "log_group_name = ",
            {
                "Ref": "WebServerLogGroup"
            },
            "\n",
            "log_stream_name = {instance_id}/apache.log\n",

            "datetime_format = %d/%b/%Y:%H:%M:%S"
        ]
    ],
    "mode": "000400",
    "owner": "apache",
    "group": "apache"
},
"/var/www/html/index.php": {
    "content": {
        "Fn::Join": [
            "",
            [
                "<?php\n",
                "echo '<h1>AWS CloudFormation sample PHP\n",

                "application</h1>';\n",

                "?>\n"
            ]
        ]
    },
    "mode": "000644",
    "owner": "apache",
    "group": "apache"
},
"/etc/cfn/cfn-hup.conf": {
    "content": {
        "Fn::Join": [
            "",
            [
                "[main]\n",
                "stack=",

```

```

        {
            "Ref": "AWS::StackId"
        },
        "\n",
        "region=",
        {
            "Ref": "AWS::Region"
        },
        "\n"
    ]
]
},
"mode": "000400",
"owner": "root",
"group": "root"
},
"/etc/cfn/hooks.d/cfn-auto-reloader.conf": {
    "content": {
        "Fn::Join": [
            "",
            [
                "[cfn-auto-reloader-hook]\n",
                "triggers=post.update\n",
                "path=Resources.WebServerHost.Metadata.AWS::CloudFormation::Init\n",
                "action=/opt/aws/bin/cfn-init -s ",
                {
                    "Ref": "AWS::StackId"
                },
                " -r WebServerHost ",
                " --region      ",
                {
                    "Ref": "AWS::Region"
                },
                "\n",
                "runas=root\n"
            ]
        ]
    }
},
"services": {
    "sysvinit": {
        "httpd": {

```

```

        "enabled": "true",
        "ensureRunning": "true"
    },
    "sendmail": {
        "enabled": "false",
        "ensureRunning": "false"
    }
}
},
"CreationPolicy": {
    "ResourceSignal": {
        "Timeout": "PT5M"
    }
},
"Properties": {
    "ImageId": {
        "Fn::FindInMap": [
            "RegionMap",
            {
                "Ref": "AWS::Region"
            }
        ],
        "AMI"
    }
},
"KeyName": {
    "Ref": "KeyName"
},
"InstanceType": "t1.micro",
"SecurityGroups": [
    {
        "Ref": "WebServerSecurityGroup"
    }
],
"IamInstanceProfile": {
    "Ref": "LogRoleInstanceProfile"
},
"UserData": {
    "Fn::Base64": {
        "Fn::Join": [
            "",
            [

```



```

        "#!/bin/bash -xe\n",
        "# Get the latest CloudFormation package\n",
        "yum install -y aws-cfn-bootstrap\n",
        "# Start cfn-init\n",
        "/opt/aws/bin/cfn-init -s ",
        {
            "Ref": "AWS::StackId"
        },
        " -r WebServerHost ",
        " --region ",
        {
            "Ref": "AWS::Region"
        },
        " || error_exit 'Failed to run cfn-init'\n",
        "# Start up the cfn-hup daemon to listen for changes to
the EC2 instance metadata\n",
        "/opt/aws/bin/cfn-hup || error_exit 'Failed to start
cfn-hup'\n",
        "# Get the CloudWatch Logs agent\n",
        "wget https://s3.amazonaws.com/aws-cloudwatch/
downloads/latest/awslogs-agent-setup.py\n",
        "# Install the CloudWatch Logs agent\n",
        "python awslogs-agent-setup.py -n -r ",
        {
            "Ref": "AWS::Region"
        },
        " -c /tmp/cwlogs/apacheaccess.conf || error_exit
'Failed to run CloudWatch Logs agent setup'\n",
        "# All done so signal success\n",
        "/opt/aws/bin/cfn-signal -e $? ",
        "      --stack ",
        {
            "Ref": "AWS::StackName"
        },
        "      --resource WebServerHost ",
        "      --region ",
        {
            "Ref": "AWS::Region"
        },
        "\n"
    ]
}
}
}

```

```

    }
  },
  "WebServerLogGroup": {
    "Type": "AWS::Logs::LogGroup",
    "Properties": {
      "RetentionInDays": 7
    }
  },
  "404MetricFilter": {
    "Type": "AWS::Logs::MetricFilter",
    "Properties": {
      "LogGroupName": {
        "Ref": "WebServerLogGroup"
      },
      "FilterPattern": "[ip, identity, user_id, timestamp, request,
status_code = 404, size, ...]",
      "MetricTransformations": [
        {
          "MetricValue": "1",
          "MetricNamespace": "test/404s",
          "MetricName": "test404Count"
        }
      ]
    }
  },
  "BytesTransferredMetricFilter": {
    "Type": "AWS::Logs::MetricFilter",
    "Properties": {
      "LogGroupName": {
        "Ref": "WebServerLogGroup"
      },
      "FilterPattern": "[ip, identity, user_id, timestamp, request,
status_code, size, ...]",
      "MetricTransformations": [
        {
          "MetricValue": "$size",
          "MetricNamespace": "test/BytesTransferred",
          "MetricName": "testBytesTransferred"
        }
      ]
    }
  },
  "404Alarm": {
    "Type": "AWS::CloudWatch::Alarm",

```

```

        "Properties": {
            "AlarmDescription": "The number of 404s is greater than 2 over 2
minutes",
            "MetricName": "test404Count",
            "Namespace": "test/404s",
            "Statistic": "Sum",
            "Period": "60",
            "EvaluationPeriods": "2",
            "Threshold": "2",
            "AlarmActions": [
                {
                    "Ref": "AlarmNotificationTopic"
                }
            ],
            "ComparisonOperator": "GreaterThanThreshold"
        }
    },
    "BandwidthAlarm": {
        "Type": "AWS::CloudWatch::Alarm",
        "Properties": {
            "AlarmDescription": "The average volume of traffic is greater 3500 KB
over 10 minutes",
            "MetricName": "testBytesTransferred",
            "Namespace": "test/BytesTransferred",
            "Statistic": "Average",
            "Period": "300",
            "EvaluationPeriods": "2",
            "Threshold": "3500",
            "AlarmActions": [
                {
                    "Ref": "AlarmNotificationTopic"
                }
            ],
            "ComparisonOperator": "GreaterThanThreshold"
        }
    },
    "AlarmNotificationTopic": {
        "Type": "AWS::SNS::Topic",
        "Properties": {
            "Subscription": [
                {
                    "Endpoint": {
                        "Ref": "OperatorEmail"
                    }
                }
            ]
        }
    }
}

```

```

        "Protocol": "email"
      }
    ]
  }
},
"Outputs": {
  "InstanceId": {
    "Description": "The instance ID of the web server",
    "Value": {
      "Ref": "WebServerHost"
    }
  },
  "WebsiteURL": {
    "Value": {
      "Fn::Join": [
        "",
        [
          "http://",
          {
            "Fn::GetAtt": [
              "WebServerHost",
              "PublicDnsName"
            ]
          }
        ]
      ]
    },
    "Description": "URL for newly created LAMP stack"
  },
  "PublicIP": {
    "Description": "Public IP address of the web server",
    "Value": {
      "Fn::GetAtt": [
        "WebServerHost",
        "PublicIp"
      ]
    }
  },
  "CloudWatchLogGroupName": {
    "Description": "The name of the CloudWatch log group",
    "Value": {
      "Ref": "WebServerLogGroup"
    }
  }
}

```

```

    }
  }
}
```

YAML

```

AWSTemplateFormatVersion: 2010-09-09
Description: AWS CloudFormation Sample Template for CloudWatch Logs.
Parameters:
  KeyName:
    Description: Name of an existing EC2 KeyPair to enable SSH access to the instances
    Type: 'AWS::EC2::KeyPair::KeyName'
    ConstraintDescription: must be the name of an existing EC2 KeyPair.
  SSHLocation:
    Description: The IP address range that can be used to SSH to the EC2 instances
    Type: String
    MinLength: '9'
    MaxLength: '18'
    Default: 0.0.0.0/0
    AllowedPattern: '(\d{1,3})\.\d{1,3})\.\d{1,3})\.\d{1,3})/(\d{1,2})'
    ConstraintDescription: must be a valid IP CIDR range of the form x.x.x.x/x.
  OperatorEmail:
    Description: Email address to notify if there are any scaling operations
    Type: String
Mappings:
  RegionMap:
    us-east-1:
      AMI: ami-0ff8a91507f77f867
    us-west-1:
      AMI: ami-0bdb828fd58c52235
    us-west-2:
      AMI: ami-a0cfeed8
    eu-west-1:
      AMI: ami-047bb4163c506cd98
    ap-southeast-1:
      AMI: ami-08569b978cc4dfa10
    ap-southeast-2:
      AMI: ami-09b42976632b27e9b
    ap-northeast-1:
      AMI: ami-06cd52961ce9f0d85
    sa-east-1:
      AMI: ami-07b14488da8ea02a0
    eu-central-1:
```

```
    AMI: ami-0233214e13e500f77
Resources:
  LogRole:
    Type: 'AWS::IAM::Role'
    Properties:
      AssumeRolePolicyDocument:
        Version: 2012-10-17
        Statement:
          - Effect: Allow
            Principal:
              Service:
                - ec2.amazonaws.com
            Action:
              - 'sts:AssumeRole'
      Path: /
      Policies:
        - PolicyName: LogRolePolicy
          PolicyDocument:
            Version: 2012-10-17
            Statement:
              - Effect: Allow
                Action:
                  - 'logs:Create*'
                  - 'logs:PutLogEvents'
                  - 's3:GetObject'
                Resource:
                  - 'arn:aws:logs:*:*:*'
                  - 'arn:aws:s3:::*'
  LogRoleInstanceProfile:
    Type: 'AWS::IAM::InstanceProfile'
    Properties:
      Path: /
      Roles:
        - !Ref LogRole
  WebServerSecurityGroup:
    Type: 'AWS::EC2::SecurityGroup'
    Properties:
      GroupDescription: Enable HTTP access via port 80 and SSH access via port 22
      SecurityGroupIngress:
        - IpProtocol: tcp
          FromPort: 80
          ToPort: 80
          CidrIp: 0.0.0.0/0
        - IpProtocol: tcp
```

```

    FromPort: 22
    ToPort: 22
    CidrIp: !Ref SSHLocation
WebServerHost:
  Type: 'AWS::EC2::Instance'
Metadata:
  Comment: Install a simple PHP application
  'AWS::CloudFormation::Init':
    config:
      packages:
        yum:
          httpd: []
          php: []
      files:
        /tmp/cwlogs/apacheaccess.conf:
          content: !Join
            - ''
            - - |
                [general]
            - |
                state_file= /var/awslogs/agent-state
            - |
                [/var/log/httpd/access_log]
            - |
                file = /var/log/httpd/access_log
            - 'log_group_name = '
            - !Ref WebServerLogGroup
            - |+

            - |
                log_stream_name = {instance_id}/apache.log
            - 'datetime_format = %d/%b/%Y:%H:%M:%S'
          mode: '000400'
          owner: apache
          group: apache
        /var/www/html/index.php:
          content: !Join
            - ''
            - - |
                <?php
            - |
                echo '<h1>AWS CloudFormation sample PHP application</h1>';
            - |
                ?>

```

```
mode: '000644'
owner: apache
group: apache
/etc/cfn/cfn-hup.conf:
content: !Join
- ''
- - |
    [main]
  - stack=
  - !Ref 'AWS::StackId'
  - |+

  - region=
  - !Ref 'AWS::Region'
  - |+

mode: '000400'
owner: root
group: root
/etc/cfn/hooks.d/cfn-auto-reloader.conf:
content: !Join
- ''
- - |
    [cfn-auto-reloader-hook]
  - |
    triggers=post.update
  - >
    path=Resources.WebServerHost.Metadata.AWS::CloudFormation::Init
  - 'action=/opt/aws/bin/cfn-init -s '
  - !Ref 'AWS::StackId'
  - ' -r WebServerHost '
  - ' --region '
  - !Ref 'AWS::Region'
  - |+

  - |
    runas=root
services:
sysvinit:
httpd:
enabled: 'true'
ensureRunning: 'true'
sendmail:
enabled: 'false'
```



```

        ensureRunning: 'false'
CreationPolicy:
  ResourceSignal:
    Timeout: PT5M
Properties:
  ImageId: !FindInMap
    - RegionMap
    - !Ref 'AWS::Region'
    - AMI
  KeyName: !Ref KeyName
  InstanceType: t1.micro
  SecurityGroups:
    - !Ref WebServerSecurityGroup
  IamInstanceProfile: !Ref LogRoleInstanceProfile
  UserData: !Base64
    'Fn::Join':
      - ''
      - - |
          #!/bin/bash -xe
        - |
          # Get the latest CloudFormation package
        - |
          yum install -y aws-cfn-bootstrap
        - |
          # Start cfn-init
        - '/opt/aws/bin/cfn-init -s '
        - !Ref 'AWS::StackId'
        - ' -r WebServerHost '
        - ' --region '
        - !Ref 'AWS::Region'
        - |2
          || error_exit 'Failed to run cfn-init'
        - >
          # Start up the cfn-hup daemon to listen for changes to the EC2
          instance metadata
        - |
          /opt/aws/bin/cfn-hup || error_exit 'Failed to start cfn-hup'
        - |
          # Get the CloudWatch Logs agent
        - >
          wget
          https://s3.amazonaws.com/aws-cloudwatch/downloads/latest/awslogs-agent-
setup.py
        - |

```

```

        # Install the CloudWatch Logs agent
        - 'python awslogs-agent-setup.py -n -r '
        - '!Ref 'AWS::Region'
        - |2
            -c /tmp/cwlogs/apacheaccess.conf || error_exit 'Failed to run CloudWatch
Logs agent setup'
        - |
            # All done so signal success
        - '/opt/aws/bin/cfn-signal -e $? '
        - '          --stack '
        - '!Ref 'AWS::StackName'
        - '          --resource WebServerHost '
        - '          --region '
        - '!Ref 'AWS::Region'
        - |+

```

WebServerLogGroup:

Type: 'AWS::Logs::LogGroup'

Properties:

RetentionInDays: 7

404MetricFilter:

Type: 'AWS::Logs::MetricFilter'

Properties:

LogGroupName: !Ref WebServerLogGroup

FilterPattern: >-

```
[ip, identity, user_id, timestamp, request, status_code = 404, size,
...]
```

MetricTransformations:

- MetricValue: '1'

MetricNamespace: test/404s

MetricName: test404Count

BytesTransferredMetricFilter:

Type: 'AWS::Logs::MetricFilter'

Properties:

LogGroupName: !Ref WebServerLogGroup

FilterPattern: '[ip, identity, user_id, timestamp, request, status_code, size, ...]'

MetricTransformations:

- MetricValue: \$size

MetricNamespace: test/BytesTransferred

MetricName: testBytesTransferred

404Alarm:

Type: 'AWS::CloudWatch::Alarm'

Properties:

```

    AlarmDescription: The number of 404s is greater than 2 over 2 minutes
    MetricName: test404Count
    Namespace: test/404s
    Statistic: Sum
    Period: '60'
    EvaluationPeriods: '2'
    Threshold: '2'
    AlarmActions:
      - !Ref AlarmNotificationTopic
    ComparisonOperator: GreaterThanThreshold
  BandwidthAlarm:
    Type: 'AWS::CloudWatch::Alarm'
    Properties:
      AlarmDescription: The average volume of traffic is greater 3500 KB over 10
minutes
      MetricName: testBytesTransferred
      Namespace: test/BytesTransferred
      Statistic: Average
      Period: '300'
      EvaluationPeriods: '2'
      Threshold: '3500'
      AlarmActions:
        - !Ref AlarmNotificationTopic
      ComparisonOperator: GreaterThanThreshold
  AlarmNotificationTopic:
    Type: 'AWS::SNS::Topic'
    Properties:
      Subscription:
        - Endpoint: !Ref OperatorEmail
          Protocol: email
  Outputs:
    InstanceId:
      Description: The instance ID of the web server
      Value: !Ref WebServerHost
    WebsiteURL:
      Value: !Join
        - ''
        - - 'http://'
          - !GetAtt
            - WebServerHost
            - PublicDnsName
      Description: URL for newly created LAMP stack
    PublicIP:
      Description: Public IP address of the web server

```

```
Value: !GetAtt
  - WebServerHost
  - PublicIp
CloudWatchLogGroupName:
  Description: The name of the CloudWatch log group
  Value: !Ref WebServerLogGroup
```

Kirim CloudWatch log ke Log dari instance Windows

Template berikut mengkonfigurasi CloudWatch Log untuk instance Windows 2012R2.

Agan CloudWatch Log di Windows (agen SSM di Windows 2012R2 dan Windows 2016 AMIs) hanya mengirim log setelah dimulai, jadi log apa pun yang dihasilkan sebelum startup tidak dikirim. Untuk mengatasi ini, templat membantu untuk memastikan agar agen dimulai sebelum log ditulis oleh:

- Pengonfigurasiannya menyiapkan agen sebagai item config yang pertama dalam configSets cfn-init.
- Penggunaan `waitAfterCompletion` untuk menyisipkan jeda setelah perintah yang memulai agen.

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "Sample template that sets up and configures CloudWatch logs on
Windows 2012R2 instance instance.",
  "Parameters": {
    "KeyPair": {
      "Description": "Name of an existing EC2 KeyPair to enable RDP access to the
instances",
      "Type": "AWS::EC2::KeyPair::KeyName",
      "ConstraintDescription": "must be the name of an existing EC2 KeyPair."
    },
    "RDPLocation": {
      "Description": "The IP address range that can be used to RDP to the EC2
instances",
      "Type": "String",
      "MinLength": "9",
      "MaxLength": "18",
      "Default": "0.0.0.0/0",
```

```
"AllowedPattern": "(\\d{1,3})\\.((\\d{1,3})|\\.|(\\d{1,3}))/(\\d{1,2})\"",
  "ConstraintDescription": "must be a valid IP CIDR range of the form x.x.x.x/x.",
},
"OperatorEmail": {
  "Description": "Email address to notify if there are any scaling operations",
  "Type": "String"
}
},
"Mappings": {
  "AWSAMIRegionMap": {
    "ap-northeast-1": {
      "WS2012R2": "ami-09e7006451ad8bf4d"
    },
    "ap-northeast-2": {
      "WS2012R2": "ami-0754980e4d02153f9"
    },
    "ap-south-1": {
      "WS2012R2": "ami-00ad91b37d56c1d08"
    },
    "ap-southeast-1": {
      "WS2012R2": "ami-09e7006451ad8bf4d"
    },
    "ap-southeast-2": {
      "WS2012R2": "ami-000d23d3067008aea"
    },
    "ca-central-1": {
      "WS2012R2": "ami-0d8e70862465b9da0"
    },
    "eu-central-1": {
      "WS2012R2": "ami-0c0f322f5676ba254"
    },
    "eu-west-1": {
      "WS2012R2": "ami-0a46adf18f8875ad6"
    },
    "eu-west-2": {
      "WS2012R2": "ami-0651428174d9438e9"
    },
    "sa-east-1": {
      "WS2012R2": "ami-08ebd138109a6c223"
    },
    "us-east-1": {
```

```

        "WS2012R2": "ami-0ef6fb504535468b2"
    },
    "us-east-2": {
        "WS2012R2": "ami-0f466c6044f510bd3"
    },
    "us-west-1": {
        "WS2012R2": "ami-026f68ef6465e6c09"
    },
    "us-west-2": {
        "WS2012R2": "ami-0274ca53943a86543"
    }
}
},
"Resources": {
    "WebServerSecurityGroup": {
        "Type": "AWS::EC2::SecurityGroup",
        "Properties": {
            "GroupDescription": "Enable HTTP access via port 80 and RDP access via
port 3389",
            "SecurityGroupIngress": [
                {
                    "IpProtocol": "tcp",
                    "FromPort": "80",
                    "ToPort": "80",
                    "CidrIp": "0.0.0.0/0"
                },
                {
                    "IpProtocol": "tcp",
                    "FromPort": "3389",
                    "ToPort": "3389",
                    "CidrIp": {
                        "Ref": "RDPLocation"
                    }
                }
            ]
        }
    },
    "LogRole": {
        "Type": "AWS::IAM::Role",
        "Properties": {
            "AssumeRolePolicyDocument": {
                "Version": "2012-10-17",
                "Statement": [
                    {

```

```

        "Effect": "Allow",
        "Principal": {
            "Service": [
                "ec2.amazonaws.com"
            ]
        },
        "Action": [
            "sts:AssumeRole"
        ]
    }
]
},
"ManagedPolicyArns": [
    "arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore"
],
"Path": "/",
"Policies": [
    {
        "PolicyName": "LogRolePolicy",
        "PolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Effect": "Allow",
                    "Action": [
                        "logs:Create*",
                        "logs:PutLogEvents",
                        "s3:GetObject"
                    ],
                    "Resource": [
                        "arn:aws:logs:*:*:*",
                        "arn:aws:s3:::*"
                    ]
                }
            ]
        }
    }
]
},
"LogRoleInstanceProfile": {
    "Type": "AWS::IAM::InstanceProfile",
    "Properties": {
        "Path": "/",

```

```

        "Roles": [
            {
                "Ref": "LogRole"
            }
        ]
    },
    "WebServerHost": {
        "Type": "AWS::EC2::Instance",
        "CreationPolicy": {
            "ResourceSignal": {
                "Timeout": "PT15M"
            }
        },
        "Metadata": {
            "AWS::CloudFormation::Init": {
                "configSets": {
                    "config": [
                        "00-ConfigureCWLogs",
                        "01-InstallWebServer",
                        "02-ConfigureApplication",
                        "03-Finalize"
                    ]
                },
                "00-ConfigureCWLogs": {
                    "files": {
                        "C:\\Program Files\\Amazon\\SSM\\Plugins\\awsCloudWatch\\
\\AWS.EC2.Windows.CloudWatch.json": {
                            "content": {
                                "Fn::Sub": "{\n  \"EngineConfiguration
\": {\n    \"Components\": [\n      {\n        \"FullName\":
\\\"AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch
\\",\n        \"Id\": \"ApplicationEventLog\", \n        \"Parameters\":
{\n          \"Levels\": \"7\", \n          \"LogName\": \"Application
\\",\n        }\n      },\n      {\n        \"FullName\":
\\\"AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch
\\",\n        \"Id\": \"SystemEventLog\", \n        \"Parameters\":
{\n          \"Levels\": \"7\", \n          \"LogName\": \"System
\\",\n        }\n      },\n      {\n        \"FullName\":
\\\"AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch
\\",\n        \"Id\": \"SecurityEventLog\", \n        \"Parameters\":
{\n          \"Levels\": \"7\", \n          \"LogName\": \"Security
\\",\n        }\n      }\n    }\n  }\n}"

```



```

\","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.
\","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

```

```

\","\\n
\\": \\""\\n
\\AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch\\",
\\n
\\Id\\": \\CloudWatchEC2ConfigLog\\",\\n
\\Parameters
\\": {\\n
\\AccessKey\\": \\",\\n
\\LogGroup\\":
\\${LogGroup}\\",\\n
\\LogStream\\": \\{instance_id}/EC2ConfigLog\\",
\\n
\\Region\\": \\${AWS::Region}\\",\\n
\\SecretKey
\\": \\""\\n
}\\n
}\\n
{\\n
\\FullName
\\": \\AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch
\\",\\n
\\Id\\": \\CloudWatchCfnInitLog\\",\\n
\\Parameters
\\": {\\n
\\AccessKey\\": \\",\\n
\\LogGroup\\":
\\${LogGroup}\\",\\n
\\LogStream\\": \\{instance_id}/CfnInitLog\\",
\\n
\\Region\\": \\${AWS::Region}\\",\\n
\\SecretKey
\\": \\""\\n
}\\n
}\\n
{\\n
\\FullName
\\": \\AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch
\\",\\n
\\Id\\": \\CloudWatchIISLogs\\",\\n
\\Parameters
\\": {\\n
\\AccessKey\\": \\",\\n
\\LogGroup\\":
\\${LogGroup}\\",\\n
\\LogStream\\": \\{instance_id}/IISLogs\\",\\n
\\Region\\": \\${AWS::Region}\\",\\n
\\SecretKey
\\": \\""\\n
}\\n
}\\n
{\\n
\\FullName\\":
\\AWS.EC2.Windows.CloudWatch.CloudWatch.CloudWatchOutputComponent,AWS.EC2.Windows.CloudWatch
\\",\\n
\\Id\\": \\CloudWatch\\",\\n
\\Parameters\\": {\\n
\\AccessKey\\": \\",\\n
\\Namespace\\": \\Windows/
Default\\",\\n
\\Region\\": \\${AWS::Region}\\",\\n
\\SecretKey\\": \\""\\n
}\\n
}\\n
]\\n
\\Flows\\": {\\n
\\Flows\\": [\\n
\\ApplicationEventLog,CloudWatchApplicationEventLog
\\",\\n
\\SystemEventLog,CloudWatchSystemEventLog\\",\\n
\\SecurityEventLog,CloudWatchSecurityEventLog\\",\\n
\\EC2ConfigLog,CloudWatchEC2ConfigLog\\",\\n
\\CfnInitLog,CloudWatchCfnInitLog\\",\\n
\\IISLogs,CloudWatchIISLogs\\",\\n
\\MemoryPerformanceCounter,CloudWatch\\""\\n
]\\n
}\\n
\\PollInterval\\": \\00:00:05\\""\\n
}\\n
\\IsEnabled\\": true\\n}\\n"
}
}
},
"commands": {
"0-enableSSM": {
"command": "powershell.exe -Command \\Set-Service -Name
AmazonSSMAgent -StartupType Automatic\\",
"waitAfterCompletion": "0"
},
"1-restartSSM": {
"command": "powershell.exe -Command \\Restart-Service
AmazonSSMAgent \\",

```

```

        "waitAfterCompletion": "30"
      }
    },
    "01-InstallWebServer": {
      "commands": {
        "01_install_webserver": {
          "command": "powershell.exe -Command \"Install-
WindowsFeature Web-Server -IncludeAllSubFeature\"",
          "waitAfterCompletion": "0"
        }
      }
    },
    "02-ConfigureApplication": {
      "files": {
        "c:\\Inetpub\\wwwroot\\index.htm": {
          "content": "<html> <head> <title>Test Application
Page</title> </head> <body> <h1>Congratulations !! Your IIS server is configured.</h1>
</body> </html>"
        }
      }
    },
    "03-Finalize": {
      "commands": {
        "00_signal_success": {
          "command": {
            "Fn::Sub": "cfn-signal.exe -e 0 --resource
WebServerHost --stack ${AWS::StackName} --region ${AWS::Region}"
          },
          "waitAfterCompletion": "0"
        }
      }
    }
  },
  "Properties": {
    "KeyName": {
      "Ref": "KeyPair"
    },
    "ImageId": {
      "Fn::FindInMap": [
        "AWSAMIRegionMap",
        {
          "Ref": "AWS::Region"
        }
      ]
    }
  }
}

```

```

        },
        "WS2012R2"
    ]
},
"InstanceType": "t2.xlarge",
"SecurityGroupIds": [
    {
        "Ref": "WebServerSecurityGroup"
    }
],
"IamInstanceProfile": {
    "Ref": "LogRoleInstanceProfile"
},
"UserData": {
    "Fn::Base64": {
        "Fn::Sub": "<script>\nwmic product where \"description='Amazon
SSM Agent' \" uninstall\nwmic product where \"description='aws-cfn-bootstrap'
\" uninstall \nstart /wait c:\\\\Windows\\\\system32\\\\msiexec /passive /qn /i
https://s3.amazonaws.com/cloudformation-examples/aws-cfn-bootstrap-win64-latest.msi
\npowershell.exe -Command \"iwr https://s3.amazonaws.com/ec2-downloads-windows/
SSMAgent/latest/windows_amd64/AmazonSSMAgentSetup.exe -UseBasicParsing -OutFile C:
\\\\AmazonSSMAgentSetup.exe\" \nstart /wait C:\\\\AmazonSSMAgentSetup.exe /install /
quiet\ncfn-init.exe -v -c config -s ${AWS::StackName} --resource WebServerHost --region
${AWS::Region} \n</script>\n"
    }
}
},
"LogGroup": {
    "Type": "AWS::Logs::LogGroup",
    "Properties": {
        "RetentionInDays": 7
    }
},
"404MetricFilter": {
    "Type": "AWS::Logs::MetricFilter",
    "Properties": {
        "LogGroupName": {
            "Ref": "LogGroup"
        },
        "FilterPattern": "[timestamps, serverip, method, uri, query, port,
dash, clientip, useragent, status_code = 404, ...]",
        "MetricTransformations": [
            {

```

```

        "MetricValue": "1",
        "MetricNamespace": "test/404s",
        "MetricName": "test404Count"
    }
]
},
"404Alarm": {
    "Type": "AWS::CloudWatch::Alarm",
    "Properties": {
        "AlarmDescription": "The number of 404s is greater than 2 over 2
minutes",
        "MetricName": "test404Count",
        "Namespace": "test/404s",
        "Statistic": "Sum",
        "Period": "60",
        "EvaluationPeriods": "2",
        "Threshold": "2",
        "AlarmActions": [
            {
                "Ref": "AlarmNotificationTopic"
            }
        ],
        "ComparisonOperator": "GreaterThanThreshold"
    }
},
"AlarmNotificationTopic": {
    "Type": "AWS::SNS::Topic",
    "Properties": {
        "Subscription": [
            {
                "Endpoint": {
                    "Ref": "OperatorEmail"
                },
                "Protocol": "email"
            }
        ]
    }
},
"Outputs": {
    "InstanceId": {
        "Description": "The instance ID of the web server",
        "Value": {

```

```

        "Ref": "WebServerHost"
      }
    },
    "WebsiteURL": {
      "Value": {
        "Fn::Sub": "http://${WebServerHost.PublicDnsName}"
      },
      "Description": "URL for newly created IIS web server"
    },
    "PublicIP": {
      "Description": "Public IP address of the web server",
      "Value": {
        "Fn::GetAtt": [
          "WebServerHost",
          "PublicIp"
        ]
      }
    },
    "CloudWatchLogGroupName": {
      "Description": "The name of the CloudWatch log group",
      "Value": {
        "Ref": "LogGroup"
      }
    }
  }
}

```

YAML

```

AWSTemplateFormatVersion: 2010-09-09
Description: >-
  Sample template that sets up and configures CloudWatch logs on Windows 2012R2
  instance instance.
Parameters:
  KeyPair:
    Description: Name of an existing EC2 KeyPair to enable RDP access to the instances
    Type: 'AWS::EC2::KeyPair::KeyName'
    ConstraintDescription: must be the name of an existing EC2 KeyPair.
  RDPLocation:
    Description: The IP address range that can be used to RDP to the EC2 instances
    Type: String
    MinLength: '9'
    MaxLength: '18'

```

```
Default: 0.0.0.0/0
AllowedPattern: '(\d{1,3})\.\d{1,3})\.\d{1,3})\.\d{1,3})/(\d{1,2})'
ConstraintDescription: must be a valid IP CIDR range of the form x.x.x.x/x.
```

OperatorEmail:

Description: Email address to notify if there are any scaling operations

Type: String

Mappings:**AWSAMIRegionMap:****ap-northeast-1:**

WS2012R2: ami-09e7006451ad8bf4d

ap-northeast-2:

WS2012R2: ami-0754980e4d02153f9

ap-south-1:

WS2012R2: ami-00ad91b37d56c1d08

ap-southeast-1:

WS2012R2: ami-09e7006451ad8bf4d

ap-southeast-2:

WS2012R2: ami-000d23d3067008aea

ca-central-1:

WS2012R2: ami-0d8e70862465b9da0

eu-central-1:

WS2012R2: ami-0c0f322f5676ba254

eu-west-1:

WS2012R2: ami-0a46adf18f8875ad6

eu-west-2:

WS2012R2: ami-0651428174d9438e9

sa-east-1:

WS2012R2: ami-08ebd138109a6c223

us-east-1:

WS2012R2: ami-0ef6fb504535468b2

us-east-2:

WS2012R2: ami-0f466c6044f510bd3

us-west-1:

WS2012R2: ami-026f68ef6465e6c09

us-west-2:

WS2012R2: ami-0274ca53943a86543

Resources:**WebServerSecurityGroup:**

Type: 'AWS::EC2::SecurityGroup'

Properties:

GroupDescription: Enable HTTP access via port 80 and RDP access via port 3389

SecurityGroupIngress:

- IpProtocol: tcp

FromPort: '80'

```
        ToPort: '80'
        CidrIp: 0.0.0.0/0
    - IpProtocol: tcp
      FromPort: '3389'
      ToPort: '3389'
      CidrIp: !Ref RDPLocation
LogRole:
  Type: 'AWS::IAM::Role'
  Properties:
    AssumeRolePolicyDocument:
      Version: 2012-10-17
      Statement:
        - Effect: Allow
          Principal:
            Service:
              - ec2.amazonaws.com
          Action:
            - 'sts:AssumeRole'
    ManagedPolicyArns:
      - 'arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore'
  Path: /
  Policies:
    - PolicyName: LogRolePolicy
      PolicyDocument:
        Version: 2012-10-17
        Statement:
          - Effect: Allow
            Action:
              - 'logs:Create*'
              - 'logs:PutLogEvents'
              - 's3:GetObject'
            Resource:
              - 'arn:aws:logs:*:*:*'
              - 'arn:aws:s3:::*'
LogRoleInstanceProfile:
  Type: 'AWS::IAM::InstanceProfile'
  Properties:
    Path: /
    Roles:
      - !Ref LogRole
WebServerHost:
  Type: 'AWS::EC2::Instance'
  CreationPolicy:
    ResourceSignal:
```



```

    Timeout: PT15M
Metadata:
  'AWS::CloudFormation::Init':
    configSets:
      config:
        - 00-ConfigureCWLogs
        - 01-InstallWebServer
        - 02-ConfigureApplication
        - 03-Finalize
    00-ConfigureCWLogs:
      files:
        'C:\Program Files\Amazon\SSM\Plugins\awsCloudWatch
\AWS.EC2.Windows.CloudWatch.json':
          content: !Sub |
            {
              "EngineConfiguration": {
                "Components": [
                  {
                    "FullName":
"\"AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch\"",
                    "Id": "ApplicationEventLog",
                    "Parameters": {
                      "Levels": "7",
                      "LogName": "Application"
                    }
                  },
                  {
                    "FullName":
"\"AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch\"",
                    "Id": "SystemEventLog",
                    "Parameters": {
                      "Levels": "7",
                      "LogName": "System"
                    }
                  },
                  {
                    "FullName":
"\"AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch\"",
                    "Id": "SecurityEventLog",
                    "Parameters": {
                      "Levels": "7",
                      "LogName": "Security"
                    }
                  }
                ]
              }
            }

```

```

        {
            "FullName":
"AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch",
            "Id": "EC2ConfigLog",
            "Parameters": {
                "CultureName": "en-US",
                "Encoding": "ASCII",
                "Filter": "EC2ConfigLog.txt",
                "LogDirectoryPath": "C:\\Program Files\\Amazon\\
\\Ec2ConfigService\\Logs",
                "TimeZoneKind": "UTC",
                "TimestampFormat": "yyyy-MM-ddTHH:mm:ss.fffZ:"
            }
        },
        {
            "FullName":
"AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch",
            "Id": "CfnInitLog",
            "Parameters": {
                "CultureName": "en-US",
                "Encoding": "ASCII",
                "Filter": "cfn-init.log",
                "LogDirectoryPath": "C:\\cfn\\log",
                "TimeZoneKind": "Local",
                "TimestampFormat": "yyyy-MM-dd HH:mm:ss,fff"
            }
        },
        {
            "FullName":
"AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch",
            "Id": "IISLogs",
            "Parameters": {
                "CultureName": "en-US",
                "Encoding": "UTF-8",
                "Filter": "",
                "LineCount": "3",
                "LogDirectoryPath": "C:\\inetpub\\logs\\LogFiles\\
\\W3SVC1",
                "TimeZoneKind": "UTC",
                "TimestampFormat": "yyyy-MM-dd HH:mm:ss"
            }
        },
        {

```

```

        "FullName":
"AWS.EC2.Windows.CloudWatch.PerformanceCounterComponent.PerformanceCounterInputComponent,AWS.E
        "Id": "MemoryPerformanceCounter",
        "Parameters": {
            "CategoryName": "Memory",
            "CounterName": "Available MBytes",
            "DimensionName": "",
            "DimensionValue": "",
            "InstanceName": "",
            "MetricName": "Memory",
            "Unit": "Megabytes"
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
        "Id": "CloudWatchApplicationEventLog",
        "Parameters": {
            "AccessKey": "",
            "LogGroup": "${LogGroup}",
            "LogStream": "{instance_id}/ApplicationEventLog",
            "Region": "${AWS::Region}",
            "SecretKey": ""
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
        "Id": "CloudWatchSystemEventLog",
        "Parameters": {
            "AccessKey": "",
            "LogGroup": "${LogGroup}",
            "LogStream": "{instance_id}/SystemEventLog",
            "Region": "${AWS::Region}",
            "SecretKey": ""
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
        "Id": "CloudWatchSecurityEventLog",
        "Parameters": {
            "AccessKey": "",
            "LogGroup": "${LogGroup}",

```

```

        "LogStream": "{instance_id}/SecurityEventLog",
        "Region": "${AWS::Region}",
        "SecretKey": ""
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
        "Id": "CloudWatchEC2ConfigLog",
        "Parameters": {
            "AccessKey": "",
            "LogGroup": "${LogGroup}",
            "LogStream": "{instance_id}/EC2ConfigLog",
            "Region": "${AWS::Region}",
            "SecretKey": ""
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
        "Id": "CloudWatchCfnInitLog",
        "Parameters": {
            "AccessKey": "",
            "LogGroup": "${LogGroup}",
            "LogStream": "{instance_id}/CfnInitLog",
            "Region": "${AWS::Region}",
            "SecretKey": ""
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
        "Id": "CloudWatchIISLogs",
        "Parameters": {
            "AccessKey": "",
            "LogGroup": "${LogGroup}",
            "LogStream": "{instance_id}/IISLogs",
            "Region": "${AWS::Region}",
            "SecretKey": ""
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CloudWatch.CloudWatchOutputComponent,AWS.EC2.Windows.CloudWatch",

```

```

        "Id": "CloudWatch",
        "Parameters": {
            "AccessKey": "",
            "NameSpace": "Windows/Default",
            "Region": "${AWS::Region}",
            "SecretKey": ""
        }
    },
    "Flows": {
        "Flows": [
            "ApplicationEventLog,CloudWatchApplicationEventLog",
            "SystemEventLog,CloudWatchSystemEventLog",
            "SecurityEventLog,CloudWatchSecurityEventLog",
            "EC2ConfigLog,CloudWatchEC2ConfigLog",
            "CfnInitLog,CloudWatchCfnInitLog",
            "IISLogs,CloudWatchIISLogs",
            "MemoryPerformanceCounter,CloudWatch"
        ]
    },
    "PollInterval": "00:00:05"
},
"IsEnabled": true
}
commands:
  0-enableSSM:
    command: >-
      powershell.exe -Command "Set-Service -Name AmazonSSMAgent
        -StartupType Automatic"
    waitAfterCompletion: '0'
  1-restartSSM:
    command: powershell.exe -Command "Restart-Service AmazonSSMAgent "
    waitAfterCompletion: '30'
01-InstallWebServer:
  commands:
    01_install_webserver:
      command: >-
        powershell.exe -Command "Install-WindowsFeature Web-Server
          -IncludeAllSubFeature"
      waitAfterCompletion: '0'
02-ConfigureApplication:
  files:
    'c:\Inetpub\wwwroot\index.htm':
      content: >-

```

```
<html> <head> <title>Test Application Page</title> </head>
<body> <h1>Congratulations !! Your IIS server is
configured.</h1> </body> </html>
```

03-Finalize:

commands:

00_signal_success:

```
command: !Sub >-
```

```
cfn-signal.exe -e 0 --resource WebServerHost --stack
```

```
${AWS::StackName} --region ${AWS::Region}
```

```
waitAfterCompletion: '0'
```

Properties:

```
KeyName: !Ref KeyPair
```

```
ImageId: !FindInMap
```

- AWSAMIRegionMap
- !Ref 'AWS::Region'
- WS2012R2

```
InstanceType: t2.xlarge
```

```
SecurityGroupIds:
```

- !Ref WebServerSecurityGroup

```
IamInstanceProfile: !Ref LogRoleInstanceProfile
```

```
UserData: !Base64
```

```
'Fn::Sub': >
```

```
<script>
```

```
wmic product where "description='Amazon SSM Agent' " uninstall
```

```
wmic product where "description='aws-cfn-bootstrap' " uninstall
```

```
start /wait c:\\Windows\\system32\\msiexec /passive /qn /i
```

```
https://s3.amazonaws.com/cloudformation-examples/aws-cfn-bootstrap-win64-
latest.msi
```

```
powershell.exe -Command "iwr
```

```
https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/latest/windows_amd64/
AmazonSSMAgentSetup.exe
```

```
-UseBasicParsing -OutFile C:\\AmazonSSMAgentSetup.exe"
```

```
start /wait C:\\AmazonSSMAgentSetup.exe /install /quiet
```

```
cfn-init.exe -v -c config -s ${AWS::StackName} --resource
```

```
WebServerHost --region ${AWS::Region}
```

```
</script>
```

```
LogGroup:
```

```
Type: 'AWS::Logs::LogGroup'
Properties:
  RetentionInDays: 7
404MetricFilter:
  Type: 'AWS::Logs::MetricFilter'
  Properties:
    LogGroupName: !Ref LogGroup
    FilterPattern: >-
      [timestamps, serverip, method, uri, query, port, dash, clientip,
       useragent, status_code = 404, ...]
    MetricTransformations:
      - MetricValue: '1'
        MetricNamespace: test/404s
        MetricName: test404Count
404Alarm:
  Type: 'AWS::CloudWatch::Alarm'
  Properties:
    AlarmDescription: The number of 404s is greater than 2 over 2 minutes
    MetricName: test404Count
    Namespace: test/404s
    Statistic: Sum
    Period: '60'
    EvaluationPeriods: '2'
    Threshold: '2'
    AlarmActions:
      - !Ref AlarmNotificationTopic
    ComparisonOperator: GreaterThanThreshold
AlarmNotificationTopic:
  Type: 'AWS::SNS::Topic'
  Properties:
    Subscription:
      - Endpoint: !Ref OperatorEmail
        Protocol: email
Outputs:
  InstanceId:
    Description: The instance ID of the web server
    Value: !Ref WebServerHost
  WebsiteURL:
    Value: !Sub 'http://${WebServerHost.PublicDnsName}'
    Description: URL for newly created IIS web server
  PublicIP:
    Description: Public IP address of the web server
    Value: !GetAtt
      - WebServerHost
```

```
- PublicIp
CloudWatchLogGroupName:
  Description: The name of the CloudWatch log group
  Value: !Ref LogGroup
```

Lihat juga

Untuk informasi selengkapnya tentang sumber daya CloudWatch Log, lihat

[AWS::Logs::LogGroup](#) atau [AWS::Logs::MetricFilter](#).

Cuplikan template Amazon DynamoDB

Topik

- [Application Auto Scaling dengan tabel Amazon DynamoDB](#)
- [Lihat juga](#)

Application Auto Scaling dengan tabel Amazon DynamoDB

Contoh ini menetapkan Application Auto Scaling untuk sumber daya `AWS::DynamoDB::Table`.

Templat tersebut menentukan kebijakan penskalaan `TargetTrackingScaling` yang menaikkan skala throughput `WriteCapacityUnits` untuk tabel.

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::GetAtt": [
            "ScalingRole",
            "Arn"
        ]
    },
    "ScalableDimension": "dynamodb:table:WriteCapacityUnits",
    "ServiceNamespace": "dynamodb"
},
"ScalingRole": {
    "Type": "AWS::IAM::Role",
    "Properties": {
        "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Effect": "Allow",
                    "Principal": {
                        "Service": [
                            "application-autoscaling.amazonaws.com"
                        ]
                    },
                    "Action": [
                        "sts:AssumeRole"
                    ]
                }
            ]
        }
    }
},
"Path": "/",
"Policies": [
    {
        "PolicyName": "root",
        "PolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [

```

```

    {
      "Effect": "Allow",
      "Action": [
        "dynamodb:DescribeTable",
        "dynamodb:UpdateTable",
        "cloudwatch:PutMetricAlarm",
        "cloudwatch:DescribeAlarms",
        "cloudwatch:GetMetricStatistics",
        "cloudwatch:SetAlarmState",
        "cloudwatch>DeleteAlarms"
      ],
      "Resource": "*"
    }
  ]
}

},
"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: !GetAtt ScalingRole.Arn
    ScalableDimension: dynamodb:table:WriteCapacityUnits
    ServiceNamespace: dynamodb
ScalingRole:
  Type: AWS::IAM::Role
  Properties:
    AssumeRolePolicyDocument:
      Version: "2012-10-17"
      Statement:
        -
          Effect: "Allow"
          Principal:
            Service:
              - application-autoscaling.amazonaws.com
          Action:
            - "sts:AssumeRole"
    Path: "/"
  Policies:
    -
      PolicyName: "root"
      PolicyDocument:
        Version: "2012-10-17"
        Statement:
          -
            Effect: "Allow"
            Action:
              - "dynamodb:DescribeTable"
              - "dynamodb:UpdateTable"
              - "cloudwatch:PutMetricAlarm"
              - "cloudwatch:DescribeAlarms"
              - "cloudwatch:GetMetricStatistics"
              - "cloudwatch:SetAlarmState"
              - "cloudwatch>DeleteAlarms"
            Resource: "*"
WriteScalingPolicy:
  Type: AWS::ApplicationAutoScaling::ScalingPolicy
  Properties:
    PolicyName: WriteAutoScalingPolicy
    PolicyType: TargetTrackingScaling
    ScalingTargetId: !Ref WriteCapacityScalableTarget
    TargetTrackingScalingPolicyConfiguration:
      TargetValue: 50.0
      ScaleInCooldown: 60
      ScaleOutCooldown: 60
```

```
PredefinedMetricSpecification:  
  PredefinedMetricType: DynamoDBWriteCapacityUtilization
```

Lihat juga

Untuk informasi selengkapnya, lihat posting blog [Cara menggunakan AWS CloudFormation untuk mengonfigurasi penskalaan otomatis untuk tabel dan indeks DynamoDB](#) di Blog Database. AWS

Untuk informasi selengkapnya tentang sumber daya DynamoDB, lihat. [AWS::DynamoDB::Table](#)

Cuplikan EC2 CloudFormation template Amazon

Amazon EC2 menyediakan kapasitas komputasi yang dapat diskalakan di. AWS Cloud Anda dapat menggunakan Amazon EC2 untuk meluncurkan server virtual sebanyak atau sesedikit yang Anda butuhkan, mengonfigurasi keamanan dan jaringan, dan mengelola penyimpanan. Server virtual ini, yang dikenal sebagai instance, dapat menjalankan berbagai sistem operasi dan aplikasi, dan dapat disesuaikan untuk memenuhi kebutuhan spesifik Anda. Amazon EC2 memungkinkan Anda meningkatkan atau menurunkan skala untuk menangani perubahan persyaratan atau lonjakan penggunaan.

Anda dapat menentukan dan menyediakan EC2 instans Amazon sebagai bagian dari infrastruktur Anda menggunakan CloudFormation templat. Template memudahkan pengelolaan dan otomatisasi penyebaran EC2 sumber daya Amazon secara berulang dan konsisten.

Contoh cuplikan template berikut menjelaskan CloudFormation sumber daya atau komponen untuk Amazon. EC2 Cuplikan ini dirancang untuk diintegrasikan ke dalam template dan tidak dimaksudkan untuk dijalankan secara independen.

Kategori Snippet

- [Konfigurasi EC2 instans Amazon dengan CloudFormation](#)
- [Buat template peluncuran dengan CloudFormation](#)
- [Kelola grup keamanan dengan CloudFormation](#)
- [Alokasikan dan kaitkan alamat IP Elastis dengan CloudFormation](#)
- [Konfigurasi sumber daya Amazon VPC dengan CloudFormation](#)

Konfigurasi EC2 instans Amazon dengan CloudFormation

Cuplikan berikut menunjukkan cara mengonfigurasi EC2 instans Amazon menggunakan CloudFormation

Kategori Snippet

- [EC2Konfigurasi Amazon umum](#)
- [Tentukan pemetaan perangkat blok untuk sebuah instance](#)

EC2Konfigurasi Amazon umum

Cuplikan berikut menunjukkan konfigurasi umum untuk instans Amazon EC2 yang digunakan CloudFormation

Contoh potongan

- [Membuat EC2 instance Amazon di Availability Zone yang ditentukan](#)
- [Mengonfigurasi EC2 instans Amazon yang diberi tag dengan volume EBS dan data pengguna](#)
- [Tentukan nama tabel DynamoDB dalam data pengguna untuk peluncuran instans Amazon EC2](#)
- [Buat volume Amazon EBS dengan DeletionPolicy](#)

Membuat EC2 instance Amazon di Availability Zone yang ditentukan

Cuplikan berikut membuat EC2 instance Amazon di Availability Zone yang ditentukan menggunakan resource. [AWS::EC2::Instance](#) Kode untuk Availability Zone adalah kode Region diikuti oleh pengenal huruf. Anda dapat meluncurkan instance ke dalam Availability Zone tunggal.

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
```

Mengonfigurasi EC2 instans Amazon yang diberi tag dengan volume EBS dan data pengguna

Cuplikan berikut membuat EC2 instance Amazon dengan tag, volume EBS, dan data pengguna. Ini menggunakan sumber [AWS::EC2::Instance](#) daya. Dalam template yang sama, Anda harus menentukan [AWS::EC2::SecurityGroup](#) sumber daya, [AWS::SNS::Topic](#) sumber daya, dan [AWS::EC2::Volume](#) sumber daya. KeyNameHarus didefinisikan di Parameters bagian template.

Tag dapat membantu Anda mengkategorikan AWS sumber daya berdasarkan preferensi Anda, seperti berdasarkan tujuan, pemilik, atau lingkungan. Data pengguna memungkinkan penyediaan skrip atau data khusus ke instance selama peluncuran. Data ini memfasilitasi otomatisasi tugas, konfigurasi perangkat lunak, instalasi paket, dan tindakan lain pada instance selama inisialisasi.

Untuk informasi selengkapnya tentang menandai sumber daya Anda, lihat [Menandai EC2 sumber daya Amazon Anda](#) di Panduan EC2 Pengguna Amazon.

Untuk informasi tentang data pengguna, lihat [Menggunakan metadata instans untuk mengelola EC2 instans Anda](#) di EC2 Panduan Pengguna Amazon.

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

```

Tentukan nama tabel DynamoDB dalam data pengguna untuk peluncuran instans Amazon EC2

Cuplikan berikut membuat EC2 instance Amazon dan mendefinisikan nama tabel DynamoDB dalam data pengguna untuk diteruskan ke instance saat peluncuran. Ini menggunakan sumber [AWS::EC2::Instance](#) daya. Anda dapat menentukan parameter atau nilai dinamis dalam data pengguna untuk meneruskan EC2 instance saat peluncuran.

Untuk informasi selengkapnya tentang data pengguna, lihat [Menggunakan metadata instans untuk mengelola EC2 instans Anda](#) di EC2 Panduan Pengguna Amazon.

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
```

Buat volume Amazon EBS dengan **DeletionPolicy**

Cuplikan berikut membuat volume Amazon EBS menggunakan sumber daya Amazon. EC2 [AWS::EC2::Volume](#) Anda dapat menggunakan SnapshotID properti Size or untuk menentukan volume, tetapi tidak keduanya. DeletionPolicyAtribut diatur untuk membuat snapshot volume saat tumpukan dihapus.

Untuk informasi selengkapnya tentang DeletionPolicy atribut, lihat [DeletionPolicyatribut](#).

Untuk informasi selengkapnya tentang membuat volume Amazon EBS, lihat [Membuat volume Amazon EBS](#).

JSON

Cuplikan ini membuat volume Amazon EBS dengan ukuran tertentu. Ukurannya diatur ke 10, tetapi Anda dapat menyesuaikannya sesuai kebutuhan. [AWS::EC2::Volume](#)Sumber daya memungkinkan Anda untuk menentukan ukuran atau ID snapshot tetapi tidak keduanya.

```
"MyEBSVolume": {
  "Type": "AWS::EC2::Volume",
  "Properties": {
    "Size": "10",
    "AvailabilityZone": {
      "Ref": "AvailabilityZone"
    }
  },
  "DeletionPolicy": "Snapshot"
}
```

Cuplikan ini membuat volume Amazon EBS menggunakan ID snapshot yang disediakan. [AWS::EC2::Volume](#)Sumber daya memungkinkan Anda untuk menentukan ukuran atau ID snapshot tetapi tidak keduanya.

```
"MyEBSVolume": {
  "Type": "AWS::EC2::Volume",
  "Properties": {
    "SnapshotId" : "snap-1234567890abcdef0",
    "AvailabilityZone": {
      "Ref": "AvailabilityZone"
    }
  },
  "DeletionPolicy": "Snapshot"
}
```

```
}
```

YAML

Cuplikan ini membuat volume Amazon EBS dengan ukuran tertentu. Ukurannya diatur ke 10, tetapi Anda dapat menyesuaikannya sesuai kebutuhan. [AWS::EC2::Volume](#) Sumber daya memungkinkan Anda untuk menentukan ukuran atau ID snapshot tetapi tidak keduanya.

```
MyEBSVolume:
  Type: AWS::EC2::Volume
  Properties:
    Size: 10
    AvailabilityZone:
      Ref: AvailabilityZone
    DeletionPolicy: Snapshot
```

Cuplikan ini membuat volume Amazon EBS menggunakan ID snapshot yang disediakan. [AWS::EC2::Volume](#) Sumber daya memungkinkan Anda untuk menentukan ukuran atau ID snapshot tetapi tidak keduanya.

```
MyEBSVolume:
  Type: AWS::EC2::Volume
  Properties:
    SnapshotId: snap-1234567890abcdef0
    AvailabilityZone:
      Ref: AvailabilityZone
    DeletionPolicy: Snapshot
```

Tentukan pemetaan perangkat blok untuk sebuah instance

Pemetaan perangkat blok mendefinisikan perangkat blok, yang mencakup volume penyimpanan instans dan volume EBS, untuk dilampirkan ke sebuah instance. Anda dapat menentukan pemetaan perangkat blok saat membuat AMI sehingga pemetaan digunakan oleh semua instance yang diluncurkan dari AMI. Atau, Anda dapat menentukan pemetaan perangkat blok saat meluncurkan instance, sehingga pemetaan menggantikan yang ditentukan dalam AMI tempat instance diluncurkan.

Anda dapat menggunakan cuplikan templat berikut untuk menentukan pemetaan perangkat blok untuk volume penyimpanan EBS atau instans menggunakan properti sumber daya.

BlockDeviceMappings [AWS::EC2::Instance](#)

Untuk informasi selengkapnya tentang memblokir pemetaan perangkat, lihat [Memblokir pemetaan perangkat untuk volume di EC2 instans Amazon di Panduan Pengguna Amazon](#). EC2

Skenario

- [Tentukan pemetaan perangkat blok untuk dua volume EBS](#)
- [Tentukan pemetaan perangkat blok untuk volume penyimpanan instance](#)

Tentukan pemetaan perangkat blok untuk dua volume EBS

JSON

```
"Ec2Instance": {
  "Type": "AWS::EC2::Instance",
  "Properties": {
    "ImageId": {
      "Fn::FindInMap": [
        "AWSRegionArch2AMI",
        { "Ref": "AWS::Region" },
        {
          "Fn::FindInMap": [
            "AWSInstanceType2Arch",
            { "Ref": "InstanceType" },
            "Arch"
          ]
        }
      ]
    },
    "KeyName": { "Ref": "KeyName" },
    "InstanceType": { "Ref": "InstanceType" },
    "SecurityGroups": [{ "Ref": "Ec2SecurityGroup" }],
    "BlockDeviceMappings": [
      {
        "DeviceName": "/dev/sda1",
        "Ebs": { "VolumeSize": "50" }
      },
      {
        "DeviceName": "/dev/sdm",
        "Ebs": { "VolumeSize": "100" }
      }
    ]
  }
}
```

```
}
```

YAML

```
EC2Instance:
  Type: AWS::EC2::Instance
  Properties:
    ImageId: !FindInMap [AWSRegionArch2AMI, !Ref 'AWS::Region', !FindInMap
[AWSInstanceType2Arch, !Ref InstanceType, Arch]]
    KeyName: !Ref KeyName
    InstanceType: !Ref InstanceType
    SecurityGroups:
      - !Ref Ec2SecurityGroup
    BlockDeviceMappings:
      -
        DeviceName: /dev/sda1
        Ebs:
          VolumeSize: 50
      -
        DeviceName: /dev/sdm
        Ebs:
          VolumeSize: 100
```

Tentukan pemetaan perangkat blok untuk volume penyimpanan instance

JSON

```
"Ec2Instance" : {
  "Type" : "AWS::EC2::Instance",
  "Properties" : {
    "ImageId" : { "Fn::FindInMap" : [ "AWSRegionArch2AMI", { "Ref" : "AWS::Region" },
"HVM64" ] },
    "KeyName" : { "Ref" : "KeyName" },
    "InstanceType": { "Ref": "InstanceType" },
    "SecurityGroups" : [{ "Ref" : "Ec2SecurityGroup" }],
    "BlockDeviceMappings" : [
      {
        "DeviceName" : "/dev/sdc",
        "VirtualName" : "ephemeral0"
      }
    ]
  }
}
```

YAML

```
EC2Instance:
  Type: AWS::EC2::Instance
  Properties:
    ImageId: !FindInMap
      - AWSRegionArch2AMI
      - !Ref AWS::Region
      - HVM64
    KeyName: !Ref KeyName
    InstanceType: !Ref InstanceType
    SecurityGroups:
      - !Ref Ec2SecurityGroup
    BlockDeviceMappings:
      - DeviceName: /dev/sdc
        VirtualName: ephemeral0
```

Buat template peluncuran dengan CloudFormation

Bagian ini memberikan contoh untuk membuat template EC2 peluncuran Amazon menggunakan CloudFormation. Template peluncuran memungkinkan Anda membuat templat untuk mengonfigurasi dan menyediakan instans EC2 Amazon di dalamnya. AWS Dengan template peluncuran, Anda dapat menyimpan parameter peluncuran sehingga Anda tidak perlu menentukannya setiap kali Anda meluncurkan instance. Untuk contoh lainnya, lihat bagian [Contoh](#) di `AWS::EC2::LaunchTemplate` sumber daya.

Untuk informasi selengkapnya tentang template peluncuran, lihat [Menyimpan parameter peluncuran instans di templat EC2 peluncuran Amazon](#) di Panduan EC2 Pengguna Amazon.

Untuk informasi tentang membuat templat peluncuran untuk digunakan dengan grup Auto Scaling, lihat templat [peluncuran Auto Scaling](#) di Panduan Pengguna Amazon Auto EC2 Scaling.

Kategori Snippet

- [Buat template peluncuran yang menentukan grup keamanan, tag, data pengguna, dan peran IAM](#)

Buat template peluncuran yang menentukan grup keamanan, tag, data pengguna, dan peran IAM

Cuplikan ini menunjukkan `AWS::EC2::LaunchTemplate` sumber daya yang berisi informasi konfigurasi untuk meluncurkan instance. Anda menentukan nilai untuk `ImageId`, `InstanceType`, `SecurityGroups`, `UserData`, dan `TagSpecifications`

properti. `SecurityGroupsProperti` menentukan grup EC2 keamanan yang ada dan grup keamanan baru. `RefFungsi` mendapatkan ID dari [AWS::EC2::SecurityGroup](#) sumber daya `myNewEC2SecurityGroup` yang dideklarasikan di tempat lain di template tumpukan.

Template peluncuran mencakup bagian untuk data pengguna kustom. Anda dapat meneruskan tugas konfigurasi dan skrip yang berjalan saat instance diluncurkan di bagian ini. Dalam contoh ini, data pengguna menginstal AWS Systems Manager Agen dan memulai agen.

Template peluncuran juga menyertakan peran IAM yang memungkinkan aplikasi yang berjalan pada instance untuk melakukan tindakan atas nama Anda. Contoh ini menunjukkan [AWS::IAM::Role](#) sumber daya untuk template peluncuran, yang menggunakan `IamInstanceProfile` properti untuk menentukan peran IAM. `RefFungsi` mendapatkan nama sumber [AWS::IAM::InstanceProfile](#) `myInstanceProfile`. Untuk mengonfigurasi izin peran IAM, Anda menentukan nilai untuk properti. `ManagedPolicyArns`

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

```

Kelola grup keamanan dengan CloudFormation

Cuplikan berikut menunjukkan cara menggunakan CloudFormation untuk mengelola grup keamanan dan EC2 instans Amazon untuk mengontrol akses ke sumber daya Anda. AWS

Kategori cuplikan

- [Kaitkan EC2 instans Amazon dengan grup keamanan](#)
- [Buat grup keamanan dengan aturan masuk](#)
- [Buat Elastic Load Balancer dengan aturan masuknya grup keamanan](#)

Kaitkan EC2 instans Amazon dengan grup keamanan

Contoh cuplikan berikut menunjukkan cara mengaitkan EC2 instance Amazon dengan grup keamanan Amazon VPC default yang digunakan. CloudFormation

Contoh potongan

- [Kaitkan EC2 instans Amazon dengan grup keamanan VPC default](#)
- [Membuat EC2 instans Amazon dengan volume terlampir dan grup keamanan](#)

Kaitkan EC2 instans Amazon dengan grup keamanan VPC default

Cuplikan berikut membuat VPC Amazon, subnet dalam VPC, dan instance Amazon. EC2 VPC dibuat menggunakan sumber daya [AWS::EC2::VPC](#). Rentang alamat IP untuk VPC didefinisikan dalam template yang lebih besar dan direferensikan oleh parameter. MyVPCCIDRRange

Subnet dibuat dalam VPC menggunakan sumber daya [AWS::EC2::Subnet](#). Subnet dikaitkan dengan VPC, yang direferensikan sebagai. MyVPC

Sebuah EC2 instance diluncurkan dalam VPC dan subnet menggunakan sumber daya. [AWS::EC2::Instance](#) Resource ini menentukan Amazon Machine Image (AMI) yang akan digunakan untuk meluncurkan instance, subnet tempat instance akan berjalan, dan grup keamanan yang akan diasosiasikan dengan instance. Fn::FindInMapFungsi ini digunakan untuk mengambil nilai dari AWSRegionToAMI pemetaan yang didefinisikan dalam template untuk menentukan sumber ImageId daya. [AWS::EC2::Instance](#)

ID grup keamanan diperoleh dengan menggunakan Fn::GetAtt fungsi, yang mengambil grup keamanan default dari MyVPC sumber daya.

Instance ditempatkan dalam MySubnet sumber daya yang ditentukan dalam cuplikan.

Saat Anda membuat VPC menggunakan CloudFormation, AWS secara otomatis membuat sumber daya default dalam VPC, termasuk grup keamanan default. Namun, ketika Anda mendefinisikan

VPC dalam CloudFormation template, Anda mungkin tidak memiliki akses ke sumber daya default ini saat Anda membuat template. IDs Untuk mengakses dan menggunakan sumber daya default yang ditentukan dalam template, Anda dapat menggunakan fungsi intrinsik seperti `Fn::GetAtt` Fungsi ini memungkinkan Anda untuk bekerja dengan sumber daya default yang dibuat secara otomatis oleh CloudFormation.

JSON

```
"MyVPC": {
  "Type": "AWS::EC2::VPC",
  "Properties": {
    "CidrBlock": {
      "Ref": "MyVPCCIDRRange"
    },
    "EnableDnsSupport": false,
    "EnableDnsHostnames": false,
    "InstanceTenancy": "default"
  }
},
"MySubnet": {
  "Type": "AWS::EC2::Subnet",
  "Properties": {
    "CidrBlock": {
      "Ref": "MyVPCCIDRRange"
    },
    "VpcId": {
      "Ref": "MyVPC"
    }
  }
},
"MyInstance": {
  "Type": "AWS::EC2::Instance",
  "Properties": {
    "ImageId": {
      "Fn::FindInMap": [
        "AWSRegionToAMI",
        {
          "Ref": "AWS::Region"
        }
      ],
      "64"
    }
  },
  "SecurityGroupIds": [
```

```

        {
            "Fn::GetAtt": [
                "MyVPC",
                "DefaultSecurityGroup"
            ]
        },
        "SubnetId": {
            "Ref": "MySubnet"
        }
    }
}

```

YAML

```

MyVPC:
  Type: AWS::EC2::VPC
  Properties:
    CidrBlock:
      Ref: MyVPCCIDRRange
    EnableDnsSupport: false
    EnableDnsHostnames: false
    InstanceTenancy: default
MySubnet:
  Type: AWS::EC2::Subnet
  Properties:
    CidrBlock:
      Ref: MyVPCCIDRRange
    VpcId:
      Ref: MyVPC
MyInstance:
  Type: AWS::EC2::Instance
  Properties:
    ImageId:
      Fn::FindInMap:
        - AWSRegionToAMI
        - Ref: AWS::Region
        - "64"
    SecurityGroupIds:
      - Fn::GetAtt:
          - MyVPC
          - DefaultSecurityGroup
    SubnetId:

```

Ref: MySubnet

Membuat EC2 instans Amazon dengan volume terlampir dan grup keamanan

Cuplikan berikut membuat EC2 instance Amazon menggunakan [AWS::EC2::Instance](#) sumber daya, yang diluncurkan dari AMI yang ditunjuk. Instance ini dikaitkan dengan grup keamanan yang memungkinkan lalu lintas SSH masuk pada port 22 dari alamat IP tertentu, menggunakan sumber daya. [AWS::EC2::SecurityGroup](#) Ini menciptakan volume Amazon EBS 100 GB menggunakan [AWS::EC2::Volume](#) sumber daya. Volume dibuat di zona ketersediaan yang sama dengan instance, seperti yang ditentukan oleh GetAtt fungsi, dan dipasang ke instance di /dev/sdh perangkat.

Untuk informasi selengkapnya tentang membuat volume Amazon EBS, lihat [Membuat volume Amazon EBS](#).

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
```

Buat grup keamanan dengan aturan masuk

Contoh cuplikan berikut menunjukkan cara mengkonfigurasi grup keamanan dengan aturan masuk tertentu menggunakan CloudFormation

Cuplikan

- [Buat grup keamanan dengan aturan ingress untuk akses SSH dan HTTP](#)
- [Buat grup keamanan dengan aturan ingress untuk akses HTTP dan SSH dari rentang CIDR yang ditentukan](#)
- [Buat grup keamanan referensi silang dengan aturan masuk](#)

Buat grup keamanan dengan aturan ingress untuk akses SSH dan HTTP

Cuplikan berikut menjelaskan dua aturan masuknya grup keamanan menggunakan sumber daya. [AWS::EC2::SecurityGroup](#) Aturan masuk pertama memungkinkan akses SSH (port 22) dari grup keamanan yang ada bernama `MyAdminSecurityGroup`, yang dimiliki oleh AWS akun dengan nomor akun. 1111-2222-3333 Aturan ingress kedua memungkinkan akses HTTP (port 80) dari grup keamanan yang berbeda bernama `MySecurityGroupCreatedInCFN`, yang dibuat dalam template yang sama. `Ref` Fungsi ini digunakan untuk referensi nama logis dari grup keamanan yang dibuat dalam template yang sama.

Dalam aturan ingress pertama, Anda harus menambahkan nilai untuk `SourceSecurityGroupOwnerId` properti `SourceSecurityGroupName` dan properti. Dalam aturan ingress kedua, `MySecurityGroupCreatedInCFN` template referensi grup keamanan yang berbeda, yang dibuat dalam template yang sama. Verifikasi bahwa nama logis `MySecurityGroupCreatedInCFN` template cocok dengan nama logis sebenarnya dari sumber daya grup keamanan yang Anda tentukan dalam templat yang lebih besar.

Untuk informasi selengkapnya tentang grup keamanan, lihat [grup EC2 keamanan Amazon untuk EC2 instans Amazon Anda](#) di Panduan EC2 Pengguna Amazon.

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
```

Buat grup keamanan dengan aturan ingress untuk akses HTTP dan SSH dari rentang CIDR yang ditentukan

Cuplikan berikut membuat grup keamanan untuk EC2 instans Amazon dengan dua aturan masuk. Aturan masuk memungkinkan lalu lintas TCP masuk pada port yang ditentukan dari rentang CIDR yang ditentukan. [AWS::EC2::SecurityGroup](#) Sumber daya digunakan untuk menentukan aturan. Anda harus menentukan protokol untuk setiap aturan. Untuk TCP, Anda harus menentukan port atau rentang port. Jika Anda tidak menentukan grup keamanan sumber atau rentang CIDR, tumpukan akan berhasil diluncurkan, tetapi aturan tidak akan diterapkan ke grup keamanan.

Untuk informasi selengkapnya tentang grup keamanan, lihat [grup EC2 keamanan Amazon untuk EC2 instans Amazon Anda](#) di Panduan EC2 Pengguna Amazon.

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

```

Buat grup keamanan referensi silang dengan aturan masuk

Cuplikan berikut menggunakan [AWS::EC2::SecurityGroup](#) sumber daya untuk membuat dua grup EC2 keamanan Amazon, SGroup1 dan SGroup2 Aturan Ingress yang memungkinkan komunikasi antara dua kelompok keamanan dibuat dengan menggunakan sumber daya [AWS::EC2::SecurityGroupIngress](#). SGroup1Ingress menetapkan aturan ingress untuk SGroup1 itu memungkinkan lalu lintas TCP masuk pada port 80 dari grup keamanan sumber,. SGroup2 SGroup2Ingress menetapkan aturan ingress untuk SGroup2 itu memungkinkan lalu lintas TCP masuk pada port 80 dari grup keamanan sumber,. 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

```

Buat Elastic Load Balancer dengan aturan masuknya grup keamanan

Template berikut membuat [AWS::ElasticLoadBalancing::LoadBalancer](#) sumber daya di zona ketersediaan yang ditentukan. [AWS::ElasticLoadBalancing::LoadBalancer](#) Sumber daya dikonfigurasi untuk mendengarkan pada port 80 untuk lalu lintas HTTP dan permintaan langsung ke instance juga pada port 80. Elastic Load Balancer bertanggung jawab untuk menyeimbangkan beban lalu lintas HTTP yang masuk di antara instans.

Selain itu, template ini menghasilkan [AWS::EC2::SecurityGroup](#) sumber daya yang terkait dengan penyeimbang beban. Grup keamanan ini dibuat dengan aturan masuk tunggal, digambarkan sebagai `ELB ingress group`, yang memungkinkan lalu lintas TCP masuk pada port 80. Sumber untuk aturan ingress ini didefinisikan menggunakan `Fn::GetAtt` function untuk mengambil atribut dari sumber daya penyeimbang beban. `SourceSecurityGroupOwnerId` digunakan `Fn::GetAtt` untuk mendapatkan kelompok keamanan sumber dari penyeimbang beban. `OwnerAlias` `SourceSecurityGroupName` digunakan `Fn::GetAtt` untuk mendapatkan kelompok keamanan sumber ELB. `GroupName`

Pengaturan ini memastikan komunikasi yang aman antara ELB dan instans.

Untuk informasi selengkapnya tentang load balancing, lihat Panduan Pengguna [Elastic Load Balancing](#).

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
```

Alokasikan dan kaitkan alamat IP Elastis dengan CloudFormation

Cuplikan template berikut adalah contoh yang terkait dengan alamat IP Elastis (EIPs) di Amazon. EC2 Contoh-contoh ini mencakup alokasi, asosiasi, dan manajemen EIPs untuk instans Anda.

Contoh potongan

- [Alokasikan alamat IP Elastis dan kaitkan dengan instans Amazon EC2](#)
- [Kaitkan alamat IP Elastis ke EC2 instans Amazon dengan menentukan alamat IP](#)
- [Kaitkan alamat IP Elastis ke EC2 instans Amazon dengan menentukan ID alokasi alamat IP](#)

Alokasikan alamat IP Elastis dan kaitkan dengan instans Amazon EC2

Cuplikan berikut mengalokasikan alamat EC2 Amazon Elastic IP (EIP) dan mengaitkannya dengan instans EC2 Amazon menggunakan sumber daya. [AWS::EC2::EIP](#) Anda dapat mengalokasikan

alamat EIP dari kumpulan alamat yang dimiliki oleh AWS atau dari kumpulan alamat yang dibuat dari rentang IPv4 alamat publik yang Anda bawa AWS untuk digunakan dengan AWS sumber daya Anda menggunakan [bawa alamat IP Anda sendiri \(BYOIP\)](#). Dalam contoh ini, EIP dialokasikan dari kumpulan alamat yang dimiliki oleh AWS.

Untuk informasi selengkapnya tentang alamat IP [Elastis](#), lihat [Alamat IP Elastis](#) di Panduan EC2 Pengguna Amazon.

JSON

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

YAML

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

Kaitkan alamat IP Elastis ke EC2 instans Amazon dengan menentukan alamat IP

Cuplikan berikut mengaitkan alamat IP Amazon EC2 Elastic yang ada ke EC2 instans menggunakan sumber daya [AWS::EC2::EIPAssociation](#). EIPAssociation Anda harus terlebih dahulu mengalokasikan alamat IP Elastis untuk digunakan di akun Anda. Alamat IP elastis dapat dikaitkan dengan satu contoh.

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
```

Kaitkan alamat IP Elastis ke EC2 instans Amazon dengan menentukan ID alokasi alamat IP

Cuplikan berikut mengaitkan alamat IP Elastic yang ada ke EC2 instans Amazon dengan menentukan ID alokasi menggunakan sumber daya [AWS::EC2::EIPAssociation](#). ID alokasi ditetapkan ke alamat IP Elastis pada alokasi alamat IP Elastis.

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
```

Konfigurasi sumber daya Amazon VPC dengan CloudFormation

Bagian ini memberikan contoh untuk mengonfigurasi sumber daya Amazon VPC menggunakan CloudFormation. VPCs memungkinkan Anda membuat jaringan virtual di dalamnya AWS, dan cuplikan ini menunjukkan cara mengonfigurasi aspek VPCs untuk memenuhi persyaratan jaringan Anda.

Contoh potongan

- [Aktifkan akses internet IPv6 khusus egress di VPC](#)
- [Potongan templat antarmuka jaringan elastis \(ENI\)](#)

Aktifkan akses internet IPv6 khusus egress di VPC

Gateway internet khusus egress memungkinkan instance dalam VPC untuk mengakses internet dan mencegah sumber daya di internet berkomunikasi dengan instans. Cuplikan berikut memungkinkan akses internet IPv6 khusus egress dari dalam VPC. Ini membuat VPC dengan rentang IPv4 alamat 10.0.0/16 menggunakan sumber daya [AWS::EC2::VPC](#). Tabel rute dikaitkan dengan sumber daya VPC ini menggunakan sumber daya [AWS::EC2::RouteTable](#). Tabel rute mengelola rute untuk instance dalam VPC. An [AWS::EC2::EgressOnlyInternetGateway](#) digunakan untuk membuat gateway internet egress-only untuk memungkinkan IPv6 komunikasi untuk lalu lintas keluar dari instance dalam VPC, sekaligus mencegah lalu lintas masuk. [AWS::EC2::Route](#) Sumber daya ditentukan untuk membuat IPv6 rute dalam tabel rute yang mengarahkan semua IPv6 lalu lintas keluar (:::/0) ke gateway internet khusus egress.

Untuk informasi selengkapnya tentang gateway internet khusus egress, lihat [Mengaktifkan IPv6 lalu lintas keluar menggunakan gateway internet khusus egress di Panduan Pengguna Amazon VPC](#).

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"

```

Potongan templat antarmuka jaringan elastis (ENI)

Buat EC2 instance Amazon dengan antarmuka jaringan elastis terlampir () ENIs

Contoh cuplikan berikut membuat EC2 instance Amazon menggunakan [AWS::EC2::Instances](#) sumber daya di VPC Amazon dan subnet yang ditentukan. Ini melampirkan dua antarmuka jaringan (ENIs) dengan instance, mengaitkan alamat IP Elastic ke instance melalui terlampir ENIs, dan mengkonfigurasi grup keamanan untuk akses SSH dan HTTP. Data pengguna diberikan ke instance sebagai bagian dari konfigurasi peluncuran saat instance dibuat. Data pengguna menyertakan skrip yang dikodekan dalam base64 format untuk memastikannya diteruskan ke instance. Ketika instance diluncurkan, skrip berjalan secara otomatis sebagai bagian dari proses bootstrap. Ini menginstal `ec2-net-utils`, mengkonfigurasi antarmuka jaringan, dan memulai layanan HTTP.

Untuk menentukan Amazon Machine Image (AMI) yang sesuai berdasarkan Wilayah yang dipilih, cuplikan menggunakan `Fn::FindInMap` fungsi yang mencari nilai dalam `RegionMap` pemetaan. Pemetaan ini harus didefinisikan dalam template yang lebih besar. Kedua antarmuka jaringan dibuat menggunakan [AWS::EC2::NetworkInterfaces](#) sumber daya. Alamat IP elastis ditentukan menggunakan [AWS::EC2::EIP](#) sumber daya yang dialokasikan ke vpc domain. Alamat IP elastis ini dikaitkan dengan antarmuka jaringan menggunakan [AWS::EC2::EIPAssociation](#) sumber daya.

Outputs Bagian ini mendefinisikan nilai atau sumber daya yang ingin Anda akses setelah tumpukan dibuat. Dalam cuplikan ini, output yang ditentukan adalah `InstancePublicIp`, yang mewakili alamat IP publik dari EC2 instance yang dibuat oleh tumpukan. Anda dapat mengambil output ini di tab Output di AWS CloudFormation konsol, atau menggunakan [perintah deskripsi-tumpukan](#).

Untuk informasi selengkapnya tentang antarmuka jaringan elastis, lihat [Antarmuka jaringan elastis](#).

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
```

Templat contoh Layanan Kontainer Elastis Amazon

Amazon Elastic Container Service (Amazon ECS) adalah layanan manajemen kontainer yang memudahkan untuk menjalankan, menghentikan, dan mengelola kontainer Docker pada sekelompok instans Amazon Elastic Compute Cloud (Amazon). EC2

Buat cluster dengan Amazon AL2 023 ECS-Optimized-AMI

Tentukan cluster yang menggunakan penyedia kapasitas yang meluncurkan AL2 023 instans di Amazon. EC2

Important

Untuk AMI terbaru IDs, lihat AMI yang [dioptimalkan Amazon ECS di Panduan](#) Pengembang Layanan Kontainer Elastis Amazon.

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}\n\n >> /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
```

Menyebarkan layanan

Template berikut mendefinisikan layanan yang menggunakan penyedia kapasitas untuk meminta kapasitas AL2 023 untuk dijalankan. Kontainer akan diluncurkan ke instans AL2 023 saat online:

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'
```

Templat Contoh Sistem File Elastis Amazon

Amazon Elastic File System (Amazon EFS) adalah layanan penyimpanan file untuk instans Amazon Elastic Compute Cloud (Amazon EC2). Dengan Amazon EFS, aplikasi Anda memiliki penyimpanan saat mereka membutuhkannya karena kapasitas penyimpanan tumbuh dan menyusut secara otomatis saat Anda menambah dan menghapus file.

Contoh template berikut menerapkan EC2 instance (dalam grup Auto Scaling) yang terkait dengan sistem file Amazon EFS. Untuk mengaitkan instans dengan sistem file, instans menjalankan skrip pembantu `cfn-init`, yang mengunduh dan menginstal paket `yum nfs-utils`, membuat direktori baru, lalu menggunakan nama DNS sistem file untuk memasang sistem file pada direktori tersebut. Nama DNS sistem file diselesaikan ke alamat IP target mount di Availability Zone EC2 instans Amazon. Untuk informasi selengkapnya tentang struktur nama DNS, lihat [Memasang Sistem File](#) di Panduan Pengguna Amazon Elastic File System.

Untuk mengukur aktivitas Sistem File Jaringan, template menyertakan CloudWatch metrik Amazon khusus. Templat juga membuat VPC, subnet, dan grup keamanan. Untuk memungkinkan instance berkomunikasi dengan sistem file, VPC harus mengaktifkan DNS, dan target pemasangan dan instance harus berada di Availability Zone (AZ) yang sama, yang ditentukan oleh subnet. EC2

Kelompok keamanan target mount memungkinkan koneksi jaringan ke port TCP 2049, yang diperlukan untuk NFSv4 klien untuk me-mount sistem file. Untuk informasi selengkapnya tentang grup keamanan untuk EC2 instans dan target pemasangan, lihat [Keamanan](#) di [Panduan Pengguna Amazon Elastic File System](#).

 Note

Jika Anda membuat pembaruan ke target pemasangan yang menyebabkannya diganti, instans atau aplikasi yang menggunakan sistem file terkait mungkin terganggu. Hal ini dapat menyebabkan penulisan tidak berkomitmen hilang. Untuk menghindari gangguan, hentikan instans Anda saat memperbarui target pemasangan dengan mengatur kapasitas yang diinginkan ke nol. Hal ini memungkinkan instans untuk membongkar sistem file sebelum target pemasangan dihapus. Setelah pembaruan pemasangan selesai, mulailah instans Anda dalam pembaruan berikutnya dengan mengatur kapasitas yang diinginkan.

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 $line | tr -s ' ' '\n`,\n",
              "      AWS_COMMAND=\"aws cloudwatch put-metric-data --region \"\n",
{ "Ref": "AWS::Region" }, "\"\n",
              "      MOD=$(( $COUNTER % 2 ))\n",
              "\n",
              "      if [ $MOD -eq 1 ]; then\n",
              "        METRIC_NAME=`echo $LINE | cut -d ' ' -f $METRIC_FIELD`\n",
\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",
\n",
"          INSTANCE_ID=$(curl -s http://169.254.169.254/latest/meta-\n",
data/instance-id)\n",
"          CW_JSON_METRIC=\"$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\n",
submit here\n",
"          aws cloudwatch put-metric-data --region ", { "Ref":\n",
"AWS::Region" }, " --cli-input-json \"`echo $CW_JSON_OPEN ${CW_JSON_METRIC%?}\n",
$CW_JSON_CLOSE`\"\n",
"\n",
"          # reset\n",
"          METRIC_COUNTER=0\n",
"          CW_JSON_METRIC=''\n",
"      fi\n",
"  fi\n",
"\n",
"\n",
"\n",
"      COUNTER=$((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":\n",
"AWS::Region" }, " --cli-input-json \"`echo $CW_JSON_OPEN ${CW_JSON_METRIC%?}\n",
$CW_JSON_CLOSE`\""\n",
" ] ] },\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,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2 ${FileSystem}.efs.
${AWS::Region}.amazonaws.com:/ /${MountPoint}" }
        },
        "02_permissions" : {
            "command" : {"Fn::Join" : [ "", [ "chown ec2-user:ec2-user /",
{ "Ref" : "MountPoint" } ] ] }
        }
    }
},
"Properties": {
    "AssociatePublicIpAddress" : true,
    "ImageId": {
        "Fn::FindInMap": [ "AWSRegionArch2AMI", { "Ref": "AWS::Region" }, {
            "Fn::FindInMap": [ "AWSInstanceType2Arch", { "Ref": "InstanceType" },
"Arch" ]
        } ]
    },
    "InstanceType": { "Ref": "InstanceType" },
    "KeyName": { "Ref": "KeyName" },
    "SecurityGroups": [ { "Ref": "InstanceSecurityGroup" } ],

```

```

    "IamInstanceProfile" : { "Ref" : "CloudWatchPutMetricsInstanceProfile" },
    "UserData"           : { "Fn::Base64" : { "Fn::Join" : [ "", [
        "#!/bin/bash -xe\n",
        "yum install -y aws-cfn-bootstrap\n",

        "/opt/aws/bin/cfn-init -v ",
        "    --stack ", { "Ref" : "AWS::StackName" },
        "    --resource LaunchConfiguration ",
        "    --configsets MountConfig ",
        "    --region ", { "Ref" : "AWS::Region" }, "\n",

        "crontab /home/ec2-user/crontab\n",

        "/opt/aws/bin/cfn-signal -e $? ",
        "    --stack ", { "Ref" : "AWS::StackName" },
        "    --resource AutoScalingGroup ",
        "    --region ", { "Ref" : "AWS::Region" }, "\n"
    ] ] ] }
  }
},
"AutoScalingGroup": {
  "Type": "AWS::AutoScaling::AutoScalingGroup",
  "DependsOn": ["MountTarget", "GatewayToInternet"],
  "CreationPolicy" : {
    "ResourceSignal" : {
      "Timeout" : "PT15M",
      "Count"   : { "Ref": "AsgMaxSize" }
    }
  },
  "Properties": {
    "VPCZoneIdentifier": [ { "Ref": "Subnet" } ],
    "LaunchConfigurationName": { "Ref": "LaunchConfiguration" },
    "MinSize": "1",
    "MaxSize": { "Ref": "AsgMaxSize" },
    "DesiredCapacity": { "Ref": "AsgMaxSize" },
    "Tags": [ {
      "Key": "Name",
      "Value": "EFS FileSystem Mounted Instance",
      "PropagateAtLaunch": "true"
    } ]
  }
},
"Outputs" : {

```

```
"MountTargetID" : {
  "Description" : "Mount target ID",
  "Value" : { "Ref" : "MountTarget" }
},
"FileSystemID" : {
  "Description" : "File system ID",
  "Value" : { "Ref" : "FileSystem" }
}
}
```

YAML

AWSTemplateFormatVersion: '2010-09-09'

Description: This template creates an Amazon EFS file system and mount target and associates it with Amazon EC2 instances in an Auto Scaling group. ****WARNING**** This template creates Amazon EC2 instances and related resources. You will be billed for the AWS resources used if you create a stack from this template.

Parameters:

InstanceType:

Description: WebServer EC2 instance type

Type: String

Default: t2.small

AllowedValues:

- t1.micro
- t2.nano
- t2.micro
- t2.small
- t2.medium
- t2.large
- m1.small
- m1.medium
- m1.large
- m1.xlarge
- m2.xlarge
- m2.2xlarge
- m2.4xlarge
- m3.medium
- m3.large
- m3.xlarge
- m3.2xlarge
- m4.large
- m4.xlarge

- m4.2xlarge
- m4.4xlarge
- m4.10xlarge
- c1.medium
- c1.xlarge
- c3.large
- c3.xlarge
- c3.2xlarge
- c3.4xlarge
- c3.8xlarge
- c4.large
- c4.xlarge
- c4.2xlarge
- c4.4xlarge
- c4.8xlarge
- g2.2xlarge
- g2.8xlarge
- r3.large
- r3.xlarge
- r3.2xlarge
- r3.4xlarge
- r3.8xlarge
- i2.xlarge
- i2.2xlarge
- i2.4xlarge
- i2.8xlarge
- d2.xlarge
- d2.2xlarge
- d2.4xlarge
- d2.8xlarge
- hi1.4xlarge
- hs1.8xlarge
- cr1.8xlarge
- cc2.8xlarge
- cg1.4xlarge

ConstraintDescription: must be a valid EC2 instance type.

KeyName:

Type: AWS::EC2::KeyPair::KeyName

Description: Name of an existing EC2 key pair to enable SSH access to the ECS instances

AsgMaxSize:

Type: Number

Description: Maximum size and initial desired capacity of Auto Scaling Group

Default: '2'

SSHLocation:

Description: The IP address range that can be used to connect to the EC2 instances by using SSH

Type: String

MinLength: '9'

MaxLength: '18'

Default: 0.0.0.0/0

AllowedPattern: "(\\d{1,3})\\. (\\d{1,3})\\. (\\d{1,3})\\. (\\d{1,3})/(\\d{1,2})"

ConstraintDescription: must be a valid IP CIDR range of the form x.x.x.x/x.

VolumeName:

Description: The name to be used for the EFS volume

Type: String

MinLength: '1'

Default: myEFSvolume

MountPoint:

Description: The Linux mount point for the EFS volume

Type: String

MinLength: '1'

Default: myEFSvolume

Mappings:

AWSInstanceType2Arch:

```
t1.micro:
```

Arch: HVM64

t2.nano:

Arch: HVM64

```
t2.micro:
```

Arch: HVM64

```
t2.small:
```

Arch: HVM64

```
t2.medium:
```

Arch: HVM64

t2.large:

Arch: HVM64

```
m1.small:
```

Arch: HVM64

```
m1.medium:
```

Arch: HVM64

```
m1.large:
```

Arch: HVM64

```
m1.xlarge:
```

Arch: HVM64

m2.xlarge:

Arch: HVM64

m2.2xlarge:

```
    Arch: HVM64
m2.4xlarge:
    Arch: HVM64
m3.medium:
    Arch: HVM64
m3.large:
    Arch: HVM64
m3.xlarge:
    Arch: HVM64
m3.2xlarge:
    Arch: HVM64
m4.large:
    Arch: HVM64
m4.xlarge:
    Arch: HVM64
m4.2xlarge:
    Arch: HVM64
m4.4xlarge:
    Arch: HVM64
m4.10xlarge:
    Arch: HVM64
c1.medium:
    Arch: HVM64
c1.xlarge:
    Arch: HVM64
c3.large:
    Arch: HVM64
c3.xlarge:
    Arch: HVM64
c3.2xlarge:
    Arch: HVM64
c3.4xlarge:
    Arch: HVM64
c3.8xlarge:
    Arch: HVM64
c4.large:
    Arch: HVM64
c4.xlarge:
    Arch: HVM64
c4.2xlarge:
    Arch: HVM64
c4.4xlarge:
    Arch: HVM64
c4.8xlarge:
```

```
    Arch: HVM64
g2.2xlarge:
    Arch: HVMG2
g2.8xlarge:
    Arch: HVMG2
r3.large:
    Arch: HVM64
r3.xlarge:
    Arch: HVM64
r3.2xlarge:
    Arch: HVM64
r3.4xlarge:
    Arch: HVM64
r3.8xlarge:
    Arch: HVM64
i2.xlarge:
    Arch: HVM64
i2.2xlarge:
    Arch: HVM64
i2.4xlarge:
    Arch: HVM64
i2.8xlarge:
    Arch: HVM64
d2.xlarge:
    Arch: HVM64
d2.2xlarge:
    Arch: HVM64
d2.4xlarge:
    Arch: HVM64
d2.8xlarge:
    Arch: HVM64
hi1.4xlarge:
    Arch: HVM64
hs1.8xlarge:
    Arch: HVM64
cr1.8xlarge:
    Arch: HVM64
cc2.8xlarge:
    Arch: HVM64
AWSRegionArch2AMI:
us-east-1:
    HVM64: ami-0ff8a91507f77f867
    HVMG2: ami-0a584ac55a7631c0c
us-west-2:
```

```
HVM64: ami-a0cfeed8
HVMG2: ami-0e09505bc235aa82d
us-west-1:
  HVM64: ami-0bdb828fd58c52235
  HVMG2: ami-066ee5fd4a9ef77f1
eu-west-1:
  HVM64: ami-047bb4163c506cd98
  HVMG2: ami-0a7c483d527806435
eu-west-2:
  HVM64: ami-f976839e
  HVMG2: NOT_SUPPORTED
eu-west-3:
  HVM64: ami-0ebc281c20e89ba4b
  HVMG2: NOT_SUPPORTED
eu-central-1:
  HVM64: ami-0233214e13e500f77
  HVMG2: ami-06223d46a6d0661c7
ap-northeast-1:
  HVM64: ami-06cd52961ce9f0d85
  HVMG2: ami-053cdd503598e4a9d
ap-northeast-2:
  HVM64: ami-0a10b2721688ce9d2
  HVMG2: NOT_SUPPORTED
ap-northeast-3:
  HVM64: ami-0d98120a9fb693f07
  HVMG2: NOT_SUPPORTED
ap-southeast-1:
  HVM64: ami-08569b978cc4dfa10
  HVMG2: ami-0be9df32ae9f92309
ap-southeast-2:
  HVM64: ami-09b42976632b27e9b
  HVMG2: ami-0a9ce9fecc3d1daf8
ap-south-1:
  HVM64: ami-0912f71e06545ad88
  HVMG2: ami-097b15e89dbdcfcf4
us-east-2:
  HVM64: ami-0b59bfac6be064b78
  HVMG2: NOT_SUPPORTED
ca-central-1:
  HVM64: ami-0b18956f
  HVMG2: NOT_SUPPORTED
sa-east-1:
  HVM64: ami-07b14488da8ea02a0
  HVMG2: NOT_SUPPORTED
```

```
cn-north-1:
  HVM64: ami-0a4eaf6c4454eda75
  HVMG2: NOT_SUPPORTED
cn-northwest-1:
  HVM64: ami-6b6a7d09
  HVMG2: NOT_SUPPORTED
Resources:
  CloudWatchPutMetricsRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Statement:
          - Effect: Allow
            Principal:
              Service:
                - ec2.amazonaws.com
            Action:
              - sts:AssumeRole
        Path: "/"
  CloudWatchPutMetricsRolePolicy:
    Type: AWS::IAM::Policy
    Properties:
      PolicyName: CloudWatch_PutMetricData
      PolicyDocument:
        Version: '2012-10-17'
        Statement:
          - Sid: CloudWatchPutMetricData
            Effect: Allow
            Action:
              - cloudwatch:PutMetricData
            Resource:
              - "*"
      Roles:
        - Ref: CloudWatchPutMetricsRole
  CloudWatchPutMetricsInstanceProfile:
    Type: AWS::IAM::InstanceProfile
    Properties:
      Path: "/"
      Roles:
        - Ref: CloudWatchPutMetricsRole
  VPC:
    Type: AWS::EC2::VPC
    Properties:
      EnableDnsSupport: 'true'
```

```
    EnableDnsHostnames: 'true'
    CidrBlock: 10.0.0.0/16
    Tags:
      - Key: Application
        Value:
          Ref: AWS::StackId
  InternetGateway:
    Type: AWS::EC2::InternetGateway
    Properties:
      Tags:
        - Key: Application
          Value:
            Ref: AWS::StackName
        - Key: Network
          Value: Public
  GatewayToInternet:
    Type: AWS::EC2::VPCGatewayAttachment
    Properties:
      VpcId:
        Ref: VPC
      InternetGatewayId:
        Ref: InternetGateway
  RouteTable:
    Type: AWS::EC2::RouteTable
    Properties:
      VpcId:
        Ref: VPC
  SubnetRouteTableAssoc:
    Type: AWS::EC2::SubnetRouteTableAssociation
    Properties:
      RouteTableId:
        Ref: RouteTable
      SubnetId:
        Ref: Subnet
  InternetGatewayRoute:
    Type: AWS::EC2::Route
    Properties:
      DestinationCidrBlock: 0.0.0.0/0
      RouteTableId:
        Ref: RouteTable
      GatewayId:
        Ref: InternetGateway
  Subnet:
    Type: AWS::EC2::Subnet
```

```
Properties:
  VpcId:
    Ref: VPC
  CidrBlock: 10.0.0.0/24
  Tags:
    - Key: Application
      Value:
        Ref: AWS::StackId
InstanceSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    VpcId:
      Ref: VPC
    GroupDescription: Enable SSH access via port 22
    SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: 22
        ToPort: 22
        CidrIp:
          Ref: SSHLocation
      - IpProtocol: tcp
        FromPort: 80
        ToPort: 80
        CidrIp: 0.0.0.0/0
MountTargetSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    VpcId:
      Ref: VPC
    GroupDescription: Security group for mount target
    SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: 2049
        ToPort: 2049
        CidrIp: 0.0.0.0/0
FileSystem:
  Type: AWS::EFS::FileSystem
  Properties:
    PerformanceMode: generalPurpose
    FileSystemTags:
      - Key: Name
        Value:
          Ref: VolumeName
MountTarget:
```



```

Type: AWS::EFS::MountTarget
Properties:
  FileSystemId:
    Ref: FileSystem
  SubnetId:
    Ref: Subnet
  SecurityGroups:
    - Ref: MountTargetSecurityGroup
LaunchConfiguration:
  Type: AWS::AutoScaling::LaunchConfiguration
  Metadata:
    AWS::CloudFormation::Init:
      configSets:
        MountConfig:
          - setup
          - mount
      setup:
        packages:
          yum:
            nfs-utils: []
        files:
          "/home/ec2-user/post_nfsstat":
            content: !Sub |
              #!/bin/bash

              INPUT="$(cat)"
              CW_JSON_OPEN='{ "Namespace": "EFS", "MetricData": [ '
              CW_JSON_CLOSE=' ] }'
              CW_JSON_METRIC=''
              METRIC_COUNTER=0

              for COL in 1 2 3 4 5 6; do

                  COUNTER=0
                  METRIC_FIELD=$COL
                  DATA_FIELD=$(( $COL + ( $COL - 1 ) ))

                  while read line; do
                      if [[ COUNTER -gt 0 ]]; then

                          LINE=`echo $line | tr -s ' ' `
                          AWS_COMMAND="aws cloudwatch put-metric-data --region
${AWS::Region}"

                          MOD=$(( $COUNTER % 2))

```

```

        if [ $MOD -eq 1 ]; then
            METRIC_NAME=`echo $LINE | cut -d ' ' -f $METRIC_FIELD`
        else
            METRIC_VALUE=`echo $LINE | cut -d ' ' -f $DATA_FIELD`
        fi

        if [[ -n "$METRIC_NAME" && -n "$METRIC_VALUE" ]]; then
            INSTANCE_ID=$(curl -s http://169.254.169.254/latest/meta-data/
instance-id)
            CW_JSON_METRIC="$CW_JSON_METRIC { \"MetricName\": \"$METRIC_NAME
\", \"Dimensions\": [{\"Name\": \"InstanceId\", \"Value\": \"$INSTANCE_ID\"} ], \"Value
\": $METRIC_VALUE },"

            unset METRIC_NAME
            unset METRIC_VALUE

            METRIC_COUNTER=$((METRIC_COUNTER+1))
            if [ $METRIC_COUNTER -eq 20 ]; then
                # 20 is max metric collection size, so we have to submit here
                aws cloudwatch put-metric-data --region ${AWS::Region} --cli-
input-json "`echo $CW_JSON_OPEN ${!CW_JSON_METRIC%?} $CW_JSON_CLOSE`"

                # reset
                METRIC_COUNTER=0
                CW_JSON_METRIC=''
            fi
        fi

        COUNTER=$((COUNTER+1))
    fi

    if [[ "$line" == "Client nfs v4:" ]]; then
        # the next line is the good stuff
        COUNTER=$((COUNTER+1))
    fi
done <<< "$INPUT"
done

# submit whatever is left
aws cloudwatch put-metric-data --region ${AWS::Region} --cli-input-json
"`echo $CW_JSON_OPEN ${!CW_JSON_METRIC%?} $CW_JSON_CLOSE`"
mode: '000755'

```

```

        owner: ec2-user
        group: ec2-user
    "/home/ec2-user/crontab":
        content: "* * * * * /usr/sbin/nfsstat | /home/ec2-user/post_nfsstat\n"
        owner: ec2-user
        group: ec2-user
    commands:
        01_createdir:
            command: !Sub "mkdir /${MountPoint}"
    mount:
        commands:
            01_mount:
                command: !Sub >
                    mount -t nfs4 -o nfsvers=4.1 ${FileSystem}.efs.
${AWS::Region}.amazonaws.com:/ /${MountPoint}
            02_permissions:
                command: !Sub "chown ec2-user:ec2-user /${MountPoint}"
Properties:
    AssociatePublicIpAddress: true
    ImageId:
        Fn::FindInMap:
            - AWSRegionArch2AMI
            - Ref: AWS::Region
        Fn::FindInMap:
            - AWSInstanceType2Arch
            - Ref: InstanceType
            - Arch
    InstanceType:
        Ref: InstanceType
    KeyName:
        Ref: KeyName
    SecurityGroups:
        - Ref: InstanceSecurityGroup
    IamInstanceProfile:
        Ref: CloudWatchPutMetricsInstanceProfile
    UserData:
        Fn::Base64: !Sub |
            #!/bin/bash -xe
            yum install -y aws-cfn-bootstrap
            /opt/aws/bin/cfn-init -v --stack ${AWS::StackName} --resource
LaunchConfiguration --configsets MountConfig --region ${AWS::Region}
            crontab /home/ec2-user/crontab
            /opt/aws/bin/cfn-signal -e $? --stack ${AWS::StackName} --resource
AutoScalingGroup --region ${AWS::Region}

```

```
AutoScalingGroup:
  Type: AWS::AutoScaling::AutoScalingGroup
  DependsOn:
    - MountTarget
    - GatewayToInternet
  CreationPolicy:
    ResourceSignal:
      Timeout: PT15M
    Count:
      Ref: AsgMaxSize
  Properties:
    VPCZoneIdentifier:
      - Ref: Subnet
    LaunchConfigurationName:
      Ref: LaunchConfiguration
    MinSize: '1'
    MaxSize:
      Ref: AsgMaxSize
    DesiredCapacity:
      Ref: AsgMaxSize
    Tags:
      - Key: Name
        Value: EFS FileSystem Mounted Instance
        PropagateAtLaunch: 'true'
  Outputs:
    MountTargetID:
      Description: Mount target ID
      Value:
        Ref: MountTarget
    FileSystemID:
      Description: File system ID
      Value:
        Ref: FileSystem
```

Cuplikan templat Elastic Beanstalk

Dengan Elastic Beanstalk, Anda dapat dengan cepat menyebarkan dan mengelola AWS aplikasi tanpa khawatir tentang infrastruktur yang menjalankan aplikasi tersebut. Contoh template berikut dapat membantu Anda menggambarkan sumber daya Elastic Beanstalk di template Anda. AWS CloudFormation

Contoh Elastic Beanstalk PHP

Contoh template berikut menyebarkan contoh aplikasi web PHP yang disimpan dalam bucket Amazon S3. Lingkungan ini juga merupakan lingkungan auto-scaling, load-balancing, dengan minimal dua instans EC2 Amazon dan maksimal enam. Ini menunjukkan lingkungan Elastic Beanstalk yang menggunakan konfigurasi peluncuran lama. Untuk selengkapnya tentang menggunakan templat peluncuran, lihat [Meluncurkan Template](#) di Panduan AWS Elastic Beanstalk Pengembang.

Ganti *solution-stack* dengan nama tumpukan penyelesaian (versi platform). Untuk daftar tumpukan solusi yang tersedia, gunakan AWS CLI perintah `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"
            ]
          }
        ]
      },
      "Description": "Beanstalk EC2 role",
      "ManagedPolicyArns": [
        "arn:aws:iam::aws:policy/AWSElasticBeanstalkWebTier",
        "arn:aws:iam::aws:policy/AWSElasticBeanstalkMulticontainerDocker",
        "arn:aws:iam::aws:policy/AWSElasticBeanstalkWorkerTier"
      ]
    }
  },
  "MyInstanceProfile": {
    "Type": "AWS::IAM::InstanceProfile",
    "Properties": {
      "Roles": [
        {
          "Ref": "MyInstanceRole"
        }
      ]
    }
  }
}

```

YAML

```
AWSTemplateFormatVersion: '2010-09-09'
Resources:
  sampleApplication:
    Type: AWS::ElasticBeanstalk::Application
    Properties:
      Description: AWS Elastic Beanstalk Sample Application
  sampleApplicationVersion:
    Type: AWS::ElasticBeanstalk::ApplicationVersion
    Properties:
      ApplicationName:
        Ref: sampleApplication
      Description: AWS ElasticBeanstalk Sample Application Version
      SourceBundle:
        S3Bucket: !Sub "elasticbeanstalk-samples-${AWS::Region}"
        S3Key: php-newsample-app.zip
  sampleConfigurationTemplate:
    Type: AWS::ElasticBeanstalk::ConfigurationTemplate
    Properties:
      ApplicationName:
        Ref: sampleApplication
      Description: AWS ElasticBeanstalk Sample Configuration Template
      OptionSettings:
        - Namespace: aws:autoscaling:asg
          OptionName: MinSize
          Value: '2'
        - Namespace: aws:autoscaling:asg
          OptionName: MaxSize
          Value: '6'
        - Namespace: aws:elasticbeanstalk:environment
          OptionName: EnvironmentType
          Value: LoadBalanced
        - Namespace: aws:autoscaling:launchconfiguration
          OptionName: IamInstanceProfile
          Value: !Ref MyInstanceProfile
      SolutionStackName: solution-stack
  sampleEnvironment:
    Type: AWS::ElasticBeanstalk::Environment
    Properties:
      ApplicationName:
        Ref: sampleApplication
      Description: AWS ElasticBeanstalk Sample Environment
      TemplateName:
```



```

    Ref: sampleConfigurationTemplate
  VersionLabel:
    Ref: sampleApplicationVersion
MyInstanceRole:
  Type: AWS::IAM::Role
  Properties:
    AssumeRolePolicyDocument:
      Version: 2012-10-17
      Statement:
        - Effect: Allow
          Principal:
            Service:
              - ec2.amazonaws.com
          Action:
            - sts:AssumeRole
    Description: Beanstalk EC2 role
    ManagedPolicyArns:
      - arn:aws:iam::aws:policy/AWSElasticBeanstalkWebTier
      - arn:aws:iam::aws:policy/AWSElasticBeanstalkMulticontainerDocker
      - arn:aws:iam::aws:policy/AWSElasticBeanstalkWorkerTier
MyInstanceProfile:
  Type: AWS::IAM::InstanceProfile
  Properties:
    Roles:
      - !Ref MyInstanceRole

```

Cuplikan templat Elastic Load Balancing

Sumber daya penyeimbang beban Elastic Load Balancing

Contoh ini menunjukkan penyeimbang beban Elastic Load Balancing dengan satu pendengar, dan tidak ada instance.

JSON

```

"MyLoadBalancer" : {
  "Type" : "AWS::ElasticLoadBalancing::LoadBalancer",
  "Properties" : {
    "AvailabilityZones" : [ "us-east-1a" ],
    "Listeners" : [ {
      "LoadBalancerPort" : "80",
      "InstancePort" : "80",
      "Protocol" : "HTTP"
    }
  ]
}

```

```
    } ]  
  }  
}
```

YAML

```
MyLoadBalancer:  
  Type: AWS::ElasticLoadBalancing::LoadBalancer  
  Properties:  
    AvailabilityZones:  
      - "us-east-1a"  
    Listeners:  
      - LoadBalancerPort: '80'  
        InstancePort: '80'  
        Protocol: HTTP
```

Sumber daya penyeimbang beban Elastic Load Balancing dengan pemeriksaan kesehatan

Contoh ini menunjukkan penyeimbang beban Elastic Load Balancing dengan dua EC2 instans Amazon, satu pendengar dan pemeriksaan kesehatan.

JSON

```
"MyLoadBalancer" : {  
  "Type" : "AWS::ElasticLoadBalancing::LoadBalancer",  
  "Properties" : {  
    "AvailabilityZones" : [ "us-east-1a" ],  
    "Instances" : [  
      { "Ref" : "logical name of AWS::EC2::Instance resource 1" },  
      { "Ref" : "logical name of AWS::EC2::Instance resource 2" }  
    ],  
    "Listeners" : [ {  
      "LoadBalancerPort" : "80",  
      "InstancePort" : "80",  
      "Protocol" : "HTTP"  
    } ],  
    "HealthCheck" : {  
      "Target" : "HTTP:80/",  
      "HealthyThreshold" : "3",  
      "UnhealthyThreshold" : "5",
```

```
        "Interval" : "30",
        "Timeout" : "5"
    }
}
```

YAML

```
MyLoadBalancer:
  Type: AWS::ElasticLoadBalancing::LoadBalancer
  Properties:
    AvailabilityZones:
      - "us-east-1a"
    Instances:
      - Ref: logical name of AWS::EC2::Instance resource 1
      - Ref: logical name of AWS::EC2::Instance resource 2
    Listeners:
      - LoadBalancerPort: '80'
        InstancePort: '80'
        Protocol: HTTP
    HealthCheck:
      Target: HTTP:80/
      HealthyThreshold: '3'
      UnhealthyThreshold: '5'
      Interval: '30'
      Timeout: '5'
```

AWS Identity and Access Management cuplikan template

Bagian ini berisi cuplikan AWS Identity and Access Management template.

Topik

- [Mendeklarasikan sumber daya pengguna IAM](#)
- [Mendeklarasikan sumber daya kunci akses IAM](#)
- [Mendeklarasikan sumber daya grup IAM](#)
- [Menambahkan pengguna ke grup](#)
- [Mendeklarasikan kebijakan IAM](#)
- [Mendeklarasikan kebijakan bucket Amazon S3](#)
- [Mendeklarasikan kebijakan topik Amazon SNS](#)

- [Mendeklarasikan kebijakan Amazon SQS](#)
- [Contoh templat IAM role](#)

Important

Saat membuat atau memperbarui tumpukan menggunakan templat yang berisi sumber daya IAM, Anda harus mengakui penggunaan kemampuan IAM. Untuk informasi selengkapnya, lihat [Mengakui sumber daya IAM dalam templat CloudFormation](#).

Mendeklarasikan sumber daya pengguna IAM

Cuplikan ini menunjukkan cara mendeklarasikan sebuah sumber daya [AWS::IAM::User](#) untuk membuat pengguna IAM. Pengguna dinyatakan dengan jalur ("/") dan profil login dengan kata sandi (myP@ssW0rd).

Dokumen kebijakan bernama `giveaccesstoqueueonly` memberikan izin kepada pengguna untuk melakukan semua tindakan Amazon SQS pada sumber daya antrian Amazon SQS, dan menolak akses ke semua `myqueue` sumber daya antrian Amazon SQS lainnya. Fungsi `Fn::GetAtt` mengambil atribut `Arn` dari `myqueue` sumber daya [AWS::SQS::Queue](#).

Dokumen kebijakan bernama `giveaccesstotopiconly` ditambahkan ke pengguna untuk memberikan izin kepada pengguna untuk melakukan semua tindakan Amazon SNS pada `mytopic` sumber daya topik Amazon SNS dan untuk menolak akses ke semua sumber daya Amazon SNS lainnya. Fungsi `Ref` mengambil ARN dari `mytopic` sumber daya [AWS::SNS::Topic](#).

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

```

Mendeklarasikan sumber daya kunci akses IAM

Cuplikan ini menunjukkan sumber daya [AWS::IAM::AccessKey](#). Sumber daya myaccesskey membuat access key dan menentukannya ke pengguna IAM yang dinyatakan sebagai sumber daya [AWS::IAM::User](#) dalam templat.

JSON

```

"myaccesskey" : {
  "Type" : "AWS::IAM::AccessKey",
  "Properties" : {
    "UserName" : { "Ref" : "myuser" }
  }
}

```

YAML

```
myaccesskey:
  Type: AWS::IAM::AccessKey
  Properties:
    UserName:
      !Ref myuser
```

Anda bisa mendapatkan kunci rahasia untuk sumber daya `AWS::IAM::AccessKey` menggunakan fungsi `Fn::GetAtt`. Salah satu cara untuk mengambil kunci rahasia adalah dengan memasukkannya ke dalam nilai `Output`. Anda bisa mendapatkan access key menggunakan fungsi `Ref`. Deklarasi nilai `Output` berikut mendapatkan access key dan kunci rahasia untuk `myaccesskey`.

JSON

```
"AccessKeyformyaccesskey" : {
  "Value" : { "Ref" : "myaccesskey" }
},
"SecretKeyformyaccesskey" : {
  "Value" : {
    "Fn::GetAtt" : [ "myaccesskey", "SecretAccessKey" ]
  }
}
```

YAML

```
AccessKeyformyaccesskey:
  Value:
    !Ref myaccesskey
SecretKeyformyaccesskey:
  Value: !GetAtt myaccesskey.SecretAccessKey
```

Anda juga dapat meneruskan kunci AWS akses dan kunci rahasia ke EC2 instans Amazon atau grup Auto Scaling yang ditentukan dalam template. Deklarasi berikut [AWS::EC2::Instance](#) menggunakan properti `UserData` untuk meneruskan access key dan kunci rahasia untuk sumber daya `myaccesskey`.

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}"

```


Mendeklarasikan sumber daya grup IAM

Cuplikan ini menunjukkan sumber daya [AWS::IAM::Group](#). Grup ini memiliki jalur ("/myapplication/"). Dokumen kebijakan bernama myapppolicy ditambahkan ke grup untuk memungkinkan pengguna grup melakukan semua tindakan Amazon SQS pada myqueue sumber daya antrian Amazon SQS dan menolak akses ke semua resource Amazon SQS lainnya kecuali myqueue

Untuk menetapkan kebijakan pada sumber daya, IAM memerlukan Amazon Resource Name (ARN) untuk sumber daya tersebut. Dalam cuplikan, fungsi Fn::GetAtt mendapat ARN dari antrean sumber daya [AWS::SQS::Queue](#).

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

```

Menambahkan pengguna ke grup

Sumber daya [AWS::IAM::UserToGroupAddition](#) menambahkan pengguna ke grup. Pada potongan berikut, sumber daya `addUserToGroup` menambahkan pengguna berikut ke grup yang ada bernama `myexistinggroup2`: pengguna yang ada `existinguser1` dan pengguna `myuser` yang dinyatakan sebagai sumber daya [AWS::IAM::User](#) dalam templat.

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
```

Mendeklarasikan kebijakan IAM

Cuplikan ini menunjukkan cara membuat kebijakan dan menerapkannya ke beberapa grup menggunakan sumber daya [AWS::IAM::Policy](#) bernama `mypolicy`. Sumber daya `mypolicy` berisi properti `PolicyDocument` yang memungkinkan tindakan `GetObject`, `PutObject`, dan `PutObjectAcl` pada objek dalam bucket S3 yang diwakili oleh `arn:aws:s3:::myAWSBucket` ARN. `mypolicy` Sumber daya menerapkan kebijakan ke grup yang ada bernama `myexistinggroup1` dan grup `mygroup` yang dideklarasikan dalam templat sebagai [AWS::IAM::Group](#) sumber daya. Contoh ini menunjukkan cara menerapkan kebijakan ke grup yang menggunakan properti `Groups`; Namun, Anda juga dapat menggunakan properti `Users` untuk menambahkan dokumen kebijakan ke daftar pengguna.

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

```

Mendeklarasikan kebijakan bucket Amazon S3

Cuplikan ini menunjukkan cara membuat kebijakan dan menerapkannya ke bucket Amazon S3 menggunakan sumber daya. [AWS::S3::BucketPolicy](#) mybucketpolicy Sumber daya mendeklarasikan dokumen kebijakan yang memungkinkan pengguna user1 IAM untuk melakukan GetObject tindakan pada semua objek dalam bucket S3 tempat kebijakan ini diterapkan. Dalam cuplikan, fungsi Fn::GetAtt mengambil ARN dari sumber daya user1. Sumber daya mybucketpolicy menerapkan kebijakan untuk my bucket sumber daya AWS::S3::BucketPolicy. Fungsi Ref mengambil nama bucket dari sumber daya mybucket.

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
```

Mendeklarasikan kebijakan topik Amazon SNS

Cuplikan ini menunjukkan cara membuat kebijakan dan menerapkannya ke topik Amazon SNS menggunakan sumber daya [AWS::SNS::TopicPolicy](#). Sumber daya `mysnspolicy` berisi properti `PolicyDocument` yang memungkinkan `myuser` sumber daya [AWS::IAM::User](#) untuk melakukan tindakan `Publish` pada `mytopic` sumber daya [AWS::SNS::Topic](#). Dalam cuplikan, fungsi `Fn::GetAtt` mendapat ARN untuk sumber daya `myuser` dan fungsi `Ref` mendapat ARN untuk sumber daya `mytopic`.

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

```

Mendeklarasikan kebijakan Amazon SQS

Cuplikan ini menunjukkan cara membuat kebijakan dan menerapkannya ke antrian Amazon SQS menggunakan sumber daya [AWS::SQS::QueuePolicy](#). Properti `PolicyDocument` memungkinkan pengguna yang ada myapp (ditentukan oleh ARN nya) untuk melakukan tindakan `SendMessage` pada antrian yang sudah ada, yang ditentukan oleh URL-nya, dan myqueue sumber daya [AWS::SQS::Queue](#). Fungsi [Ref](#) mendapat URL untuk sumber daya myqueue.

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

```

Contoh templat IAM role

Bagian ini memberikan contoh CloudFormation template untuk peran IAM untuk EC2 instance.

Untuk informasi selengkapnya, lihat [peran IAM untuk Amazon EC2](#) di Panduan EC2 Pengguna Amazon.

Peran IAM dengan EC2

Dalam contoh ini, profil instance direferensikan oleh `IamInstanceProfile` properti EC2 instance. Kebijakan instans dan referensi kebijakan peran [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

```

Peran IAM dengan grup Auto Scaling

Dalam contoh ini, profil instance direferensikan oleh `IamInstanceProfile` properti konfigurasi peluncuran Amazon EC2 Auto Scaling.

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 Template

Template berikut menggunakan fungsi AWS Lambda (Lambda) dan sumber daya khusus untuk menambahkan grup keamanan baru ke daftar grup keamanan yang ada. Fungsi ini berguna ketika Anda ingin membangun daftar grup keamanan secara dinamis, sehingga daftar Anda mencakup grup keamanan baru dan yang sudah ada. Misalnya, Anda dapat meneruskan daftar grup keamanan yang ada sebagai nilai parameter, menambahkan nilai baru ke daftar, dan kemudian mengaitkan semua nilai Anda dengan sebuah EC2 instance. Untuk informasi selengkapnya tentang jenis sumber daya fungsi Lambda, lihat [AWS::Lambda::Function](#)

Dalam contoh, saat CloudFormation membuat sumber daya AllSecurityGroups khusus, CloudFormation memanggil fungsi AppendItemToListFunction Lambda. CloudFormation meneruskan daftar grup keamanan yang ada dan grup keamanan baru (NewSecurityGroup) ke fungsi, yang menambahkan grup keamanan baru ke daftar dan kemudian mengembalikan daftar

yang dimodifikasi. CloudFormation menggunakan daftar yang dimodifikasi untuk mengaitkan semua grup keamanan dengan MyEC2Instance sumber daya.

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Parameters": {
    "ExistingSecurityGroups": {
      "Type": "List<AWS::EC2::SecurityGroup::Id>"
    },
    "ExistingVPC": {
      "Type": "AWS::EC2::VPC::Id",
      "Description": "The VPC ID that includes the security groups in the ExistingSecurityGroups parameter."
    },
    "InstanceType": {
      "Type": "String",
      "Default": "t2.micro",
      "AllowedValues": [
        "t2.micro",
        "m1.small"
      ]
    }
  },
  "Mappings": {
    "AWSInstanceType2Arch": {
      "t2.micro": {
        "Arch": "HVM64"
      },
      "m1.small": {
        "Arch": "HVM64"
      }
    },
    "AWSRegionArch2AMI": {
      "us-east-1": {
        "HVM64": "ami-0ff8a91507f77f867",
        "HVMG2": "ami-0a584ac55a7631c0c"
      },
      "us-west-2": {
        "HVM64": "ami-a0cfeed8",
        "HVMG2": "ami-0e09505bc235aa82d"
      }
    }
  }
}
```

```

    "us-west-1": {
        "HVM64": "ami-0bdb828fd58c52235",
        "HVMG2": "ami-066ee5fd4a9ef77f1"
    },
    "eu-west-1": {
        "HVM64": "ami-047bb4163c506cd98",
        "HVMG2": "ami-0a7c483d527806435"
    },
    "eu-central-1": {
        "HVM64": "ami-0233214e13e500f77",
        "HVMG2": "ami-06223d46a6d0661c7"
    },
    "ap-northeast-1": {
        "HVM64": "ami-06cd52961ce9f0d85",
        "HVMG2": "ami-053cdd503598e4a9d"
    },
    "ap-southeast-1": {
        "HVM64": "ami-08569b978cc4dfa10",
        "HVMG2": "ami-0be9df32ae9f92309"
    },
    "ap-southeast-2": {
        "HVM64": "ami-09b42976632b27e9b",
        "HVMG2": "ami-0a9ce9fecc3d1daf8"
    },
    "sa-east-1": {
        "HVM64": "ami-07b14488da8ea02a0",
        "HVMG2": "NOT_SUPPORTED"
    },
    "cn-north-1": {
        "HVM64": "ami-0a4eaf6c4454eda75",
        "HVMG2": "NOT_SUPPORTED"
    }
}
},
"Resources": {
    "SecurityGroup": {
        "Type": "AWS::EC2::SecurityGroup",
        "Properties": {
            "GroupDescription": "Allow HTTP traffic to the host",
            "VpcId": {
                "Ref": "ExistingVPC"
            },
        },
        "SecurityGroupIngress": [
            {

```

```

        "IpProtocol": "tcp",
        "FromPort": 80,
        "ToPort": 80,
        "CidrIp": "0.0.0.0/0"
    }
],
"SecurityGroupEgress": [
    {
        "IpProtocol": "tcp",
        "FromPort": 80,
        "ToPort": 80,
        "CidrIp": "0.0.0.0/0"
    }
]
},
"AllSecurityGroups": {
    "Type": "Custom::Split",
    "Properties": {
        "ServiceToken": {
            "Fn::GetAtt": [
                "AppendItemToListFunction",
                "Arn"
            ]
        },
        "List": {
            "Ref": "ExistingSecurityGroups"
        },
        "AppendedItem": {
            "Ref": "SecurityGroup"
        }
    }
},
"AppendItemToListFunction": {
    "Type": "AWS::Lambda::Function",
    "Properties": {
        "Handler": "index.handler",
        "Role": {
            "Fn::GetAtt": [
                "LambdaExecutionRole",
                "Arn"
            ]
        },
        "Code": {

```



```

        "ZipFile": {
            "Fn::Join": [
                "",
                [
                    "var response = require('cfn-response');",
                    "exports.handler = function(event, context) {",
                    "    var responseData = {Value:",
event.ResourceProperties.List};",
                    "    ",
                    "    responseData.Value.push(event.ResourceProperties.AppendedItem);",
                    "    response.send(event, context, response.SUCCESS,",
                    responseData);",
                    "};"
                ]
            ]
        },
        "Runtime": "nodejs20.x"
    },
    "MyEC2Instance": {
        "Type": "AWS::EC2::Instance",
        "Properties": {
            "ImageId": {
                "Fn::FindInMap": [
                    "AWSRegionArch2AMI",
                    {
                        "Ref": "AWS::Region"
                    }
                ],
                {
                    "Fn::FindInMap": [
                        "AWSInstanceType2Arch",
                        {
                            "Ref": "InstanceType"
                        },
                        "Arch"
                    ]
                }
            ]
        }
    },
    "SecurityGroupIds": {
        "Fn::GetAtt": [
            "AllSecurityGroups",
            "Value"
        ]
    }
}

```

```

        ]
      },
      "InstanceType": {
        "Ref": "InstanceType"
      }
    }
  },
  "LambdaExecutionRole": {
    "Type": "AWS::IAM::Role",
    "Properties": {
      "AssumeRolePolicyDocument": {
        "Version": "2012-10-17",
        "Statement": [
          {
            "Effect": "Allow",
            "Principal": {
              "Service": [
                "lambda.amazonaws.com"
              ]
            },
            "Action": [
              "sts:AssumeRole"
            ]
          }
        ]
      }
    }
  },
  "Path": "/",
  "Policies": [
    {
      "PolicyName": "root",
      "PolicyDocument": {
        "Version": "2012-10-17",
        "Statement": [
          {
            "Effect": "Allow",
            "Action": [
              "logs:*"
            ],
            "Resource": "arn:aws:logs:*:*:*"
          }
        ]
      }
    }
  ]
}

```

```

    }
  }
},
"Outputs": {
  "AllSecurityGroups": {
    "Description": "Security Groups that are associated with the EC2 instance",
    "Value": {
      "Fn::Join": [
        ",",
        [
          {
            "Fn::GetAtt": [
              "AllSecurityGroups",
              "Value"
            ]
          }
        ]
      ]
    }
  }
}
}

```

YAML

```

AWSTemplateFormatVersion: '2010-09-09'
Parameters:
  ExistingSecurityGroups:
    Type: List<AWS::EC2::SecurityGroup::Id>
  ExistingVPC:
    Type: AWS::EC2::VPC::Id
    Description: The VPC ID that includes the security groups in the
ExistingSecurityGroups parameter.
  InstanceType:
    Type: String
    Default: t2.micro
    AllowedValues:
      - t2.micro
      - m1.small
Mappings:
  AWSInstanceType2Arch:
    t2.micro:
      Arch: HVM64
    m1.small:
      Arch: HVM64

```

AWSRegionArch2AMI:**us-east-1:**

HVM64: ami-0ff8a91507f77f867

HVMG2: ami-0a584ac55a7631c0c

us-west-2:

HVM64: ami-a0cfeed8

HVMG2: ami-0e09505bc235aa82d

us-west-1:

HVM64: ami-0bdb828fd58c52235

HVMG2: ami-066ee5fd4a9ef77f1

eu-west-1:

HVM64: ami-047bb4163c506cd98

HVMG2: ami-0a7c483d527806435

eu-central-1:

HVM64: ami-0233214e13e500f77

HVMG2: ami-06223d46a6d0661c7

ap-northeast-1:

HVM64: ami-06cd52961ce9f0d85

HVMG2: ami-053cdd503598e4a9d

ap-southeast-1:

HVM64: ami-08569b978cc4dfa10

HVMG2: ami-0be9df32ae9f92309

ap-southeast-2:

HVM64: ami-09b42976632b27e9b

HVMG2: ami-0a9ce9fecc3d1daf8

sa-east-1:

HVM64: ami-07b14488da8ea02a0

HVMG2: NOT_SUPPORTED

cn-north-1:

HVM64: ami-0a4eaf6c4454eda75

HVMG2: NOT_SUPPORTED

Resources:**SecurityGroup:**

Type: AWS::EC2::SecurityGroup

Properties:

GroupDescription: Allow HTTP traffic to the host

VpcId: !Ref ExistingVPC

SecurityGroupIngress:

- IpProtocol: tcp

FromPort: 80

ToPort: 80

CidrIp: 0.0.0.0/0

SecurityGroupEgress:

- IpProtocol: tcp

```

        FromPort: 80
        ToPort: 80
        CidrIp: 0.0.0.0/0
AllSecurityGroups:
  Type: Custom::Split
  Properties:
    ServiceToken: !GetAtt AppendItemToListFunction.Arn
    List: !Ref ExistingSecurityGroups
    AppendedItem: !Ref SecurityGroup
AppendItemToListFunction:
  Type: AWS::Lambda::Function
  Properties:
    Handler: index.handler
    Role: !GetAtt LambdaExecutionRole.Arn
    Code:
      ZipFile: !Join
        - ''
        - - var response = require('cfn-response');
          - exports.handler = function(event, context) {
            - '  var responseData = {Value: event.ResourceProperties.List};'
            - '    responseData.Value.push(event.ResourceProperties.AppendedItem);'
            - '    response.send(event, context, response.SUCCESS, responseData);'
            - '};'
    Runtime: nodejs20.x
MyEC2Instance:
  Type: AWS::EC2::Instance
  Properties:
    ImageId: !FindInMap
      - AWSRegionArch2AMI
      - !Ref AWS::Region
      - !FindInMap
        - AWSInstanceType2Arch
        - !Ref InstanceType
        - Arch
    SecurityGroupIds: !GetAtt AllSecurityGroups.Value
    InstanceType: !Ref InstanceType
LambdaExecutionRole:
  Type: AWS::IAM::Role
  Properties:
    AssumeRolePolicyDocument:
      Version: '2012-10-17'
      Statement:
        - Effect: Allow
          Principal:

```

```
    Service:
      - lambda.amazonaws.com
    Action:
      - sts:AssumeRole
  Path: /
  Policies:
    - PolicyName: root
      PolicyDocument:
        Version: '2012-10-17'
        Statement:
          - Effect: Allow
            Action:
              - logs:*
            Resource: arn:aws:logs:*:*:*
  Outputs:
    AllSecurityGroups:
      Description: Security Groups that are associated with the EC2 instance
      Value: !Join
        - ', '
        - !GetAtt AllSecurityGroups.Value
```

AWS OpsWorks cuplikan template

AWS OpsWorks adalah layanan manajemen aplikasi yang menyederhanakan berbagai tugas seperti konfigurasi perangkat lunak, penerapan aplikasi, penskalaan, dan pemantauan. CloudFormation adalah layanan manajemen sumber daya yang dapat Anda gunakan untuk mengelola AWS OpsWorks sumber daya, seperti AWS OpsWorks tumpukan, lapisan, aplikasi, dan instance.

AWS OpsWorks contoh aplikasi PHP

Contoh template berikut menyebarkan contoh aplikasi web AWS OpsWorks PHP yang disimpan dalam repositori Git publik. AWS OpsWorks Tumpukan mencakup dua server aplikasi dengan penyeimbang beban yang mendistribusikan lalu lintas masuk secara merata di seluruh server. AWS OpsWorks Tumpukan juga mencakup server database MySQL back-end untuk menyimpan data. Untuk informasi selengkapnya tentang contoh AWS OpsWorks aplikasi, lihat [Panduan: Pelajari AWS OpsWorks dasar-dasar dengan membuat tumpukan server aplikasi](#) di AWS OpsWorks Panduan Pengguna.

 Note

Referensi `ServiceRoleArn` dan `DefaultInstanceProfileArn` properti peran IAM yang dibuat setelah Anda gunakan AWS OpsWorks untuk pertama kalinya.

Contoh mendefinisikan parameter `MysqlRootPassword` dengan properti `NoEcho` diatur menjadi `true`. Jika Anda mengatur atribut `NoEcho` menjadi `true`, CloudFormation mengembalikan nilai parameter tertutup sebagai tanda bintang (`*****`) untuk setiap panggilan yang menggambarkan tumpukan atau peristiwa tumpukan.

 Important

Daripada menyematkan informasi sensitif secara langsung di CloudFormation template Anda, kami sarankan Anda menggunakan parameter dinamis dalam template tumpukan untuk mereferensikan informasi sensitif yang disimpan dan dikelola di luar CloudFormation, seperti di AWS Systems Manager Parameter Store atau AWS Secrets Manager.

Untuk informasi selengkapnya, lihat [Do not embed credentials dalam praktik terbaik template Anda](#).

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Parameters": {
    "ServiceRole": {
      "Default": "aws-opsworks-service-role",
      "Description": "The OpsWorks service role",
      "Type": "String",
      "MinLength": "1",
      "MaxLength": "64",
      "AllowedPattern": "[a-zA-Z][a-zA-Z0-9]*",
      "ConstraintDescription": "must begin with a letter and contain only alphanumeric characters."
    },
    "InstanceRole": {
      "Default": "aws-opsworks-ec2-role",
      "Description": "The OpsWorks instance role",
      "Type": "String",
```

```

    "MinLength": "1",
    "MaxLength": "64",
    "AllowedPattern": "[a-zA-Z][a-zA-Z0-9]*",
    "ConstraintDescription": "must begin with a letter and contain only alphanumeric
characters."
  },
  "AppName": {
    "Default": "myapp",
    "Description": "The app name",
    "Type": "String",
    "MinLength": "1",
    "MaxLength": "64",
    "AllowedPattern": "[a-zA-Z][a-zA-Z0-9]*",
    "ConstraintDescription": "must begin with a letter and contain only alphanumeric
characters."
  },
  "MysqlRootPassword" : {
    "Description" : "MysqlRootPassword",
    "NoEcho" : "true",
    "Type" : "String"
  }
},
"Resources": {
  "myStack": {
    "Type": "AWS::OpsWorks::Stack",
    "Properties": {
      "Name": {
        "Ref": "AWS::StackName"
      },
      "ServiceRoleArn": {
        "Fn::Join": [
          "", ["arn:aws:iam::", {"Ref": "AWS::AccountId"},
            ":role/", {"Ref": "ServiceRole"}]
        ]
      },
      "DefaultInstanceProfileArn": {
        "Fn::Join": [
          "", ["arn:aws:iam::", {"Ref": "AWS::AccountId"},
            ":instance-profile/", {"Ref": "InstanceRole"}]
        ]
      },
      "UseCustomCookbooks": "true",
      "CustomCookbooksSource": {
        "Type": "git",

```



```

        "Url": "git://github.com/amazonwebservices/opsworks-example-cookbooks.git"
    }
}
},
"myLayer": {
    "Type": "AWS::OpsWorks::Layer",
    "DependsOn": "myApp",
    "Properties": {
        "StackId": {"Ref": "myStack"},
        "Type": "php-app",
    "Shortname" : "php-app",
        "EnableAutoHealing" : "true",
        "AutoAssignElasticIps" : "false",
        "AutoAssignPublicIps" : "true",
        "Name": "MyPHPApp",
        "CustomRecipes" : {
            "Configure" : ["phpapp::appsetup"]
        }
    }
},
"DBLayer" : {
    "Type" : "AWS::OpsWorks::Layer",
    "DependsOn": "myApp",
    "Properties" : {
        "StackId" : {"Ref":"myStack"},
        "Type" : "db-master",
    "Shortname" : "db-layer",
        "EnableAutoHealing" : "true",
        "AutoAssignElasticIps" : "false",
        "AutoAssignPublicIps" : "true",
        "Name" : "MyMySQL",
        "CustomRecipes" : {
            "Setup" : ["phpapp::dbsetup"]
        },
        "Attributes" : {
            "MysqlRootPassword" : {"Ref":"MysqlRootPassword"},
            "MysqlRootPasswordUbiquitous": "true"
        },
        "VolumeConfigurations":[{"MountPoint":"/vol/
mysql","NumberOfDisks":1,"Size":10}]
    }
},
"ELBAttachment" : {
    "Type" : "AWS::OpsWorks::ElasticLoadBalancerAttachment",

```

```
"Properties" : {
  "ElasticLoadBalancerName" : { "Ref" : "ELB" },
  "LayerId" : { "Ref" : "myLayer" }
},
"ELB" : {
  "Type": "AWS::ElasticLoadBalancing::LoadBalancer",
  "Properties": {
    "AvailabilityZones": { "Fn::GetAZs" : "" } ,
    "Listeners": [{
      "LoadBalancerPort": "80",
      "InstancePort": "80",
      "Protocol": "HTTP",
      "InstanceProtocol": "HTTP"
    }],
    "HealthCheck": {
      "Target": "HTTP:80/",
      "HealthyThreshold": "2",
      "UnhealthyThreshold": "10",
      "Interval": "30",
      "Timeout": "5"
    }
  }
},
"myAppInstance1": {
  "Type": "AWS::OpsWorks::Instance",
  "Properties": {
    "StackId": {"Ref": "myStack"},
    "LayerIds": [{"Ref": "myLayer"}],
    "InstanceType": "m1.small"
  }
},
"myAppInstance2": {
  "Type": "AWS::OpsWorks::Instance",
  "Properties": {
    "StackId": {"Ref": "myStack"},
    "LayerIds": [{"Ref": "myLayer"}],
    "InstanceType": "m1.small"
  }
},
"myDBInstance": {
  "Type": "AWS::OpsWorks::Instance",
  "Properties": {
    "StackId": {"Ref": "myStack"},
```

```

        "LayerIds": [{"Ref": "DBLayer"}],
        "InstanceType": "m1.small"
    }
},
"myApp" : {
    "Type" : "AWS::OpsWorks::App",
    "Properties" : {
        "StackId" : {"Ref": "myStack"},
        "Type" : "php",
        "Name" : {"Ref": "AppName"},
        "AppSource" : {
            "Type" : "git",
            "Url" : "git://github.com/amazonwebservices/opsworks-demo-php-simple-
app.git",
            "Revision" : "version2"
        },
        "Attributes" : {
            "DocumentRoot" : "web"
        }
    }
}
}
}
}

```

YAML

```

AWSTemplateFormatVersion: '2010-09-09'
Parameters:
  ServiceRole:
    Default: aws-opsworks-service-role
    Description: The OpsWorks service role
    Type: String
    MinLength: '1'
    MaxLength: '64'
    AllowedPattern: "[a-zA-Z][a-zA-Z0-9-]*"
    ConstraintDescription: must begin with a letter and contain only alphanumeric
      characters.
  InstanceRole:
    Default: aws-opsworks-ec2-role
    Description: The OpsWorks instance role
    Type: String
    MinLength: '1'
    MaxLength: '64'

```

```
    AllowedPattern: "[a-zA-Z][a-zA-Z0-9-]*"
    ConstraintDescription: must begin with a letter and contain only alphanumeric
        characters.
  AppName:
    Default: myapp
    Description: The app name
    Type: String
    MinLength: '1'
    MaxLength: '64'
    AllowedPattern: "[a-zA-Z][a-zA-Z0-9-]*"
    ConstraintDescription: must begin with a letter and contain only alphanumeric
        characters.
  MysqlRootPassword:
    Description: MysqlRootPassword
    NoEcho: 'true'
    Type: String
Resources:
  myStack:
    Type: AWS::OpsWorks::Stack
    Properties:
      Name:
        Ref: AWS::StackName
      ServiceRoleArn: !Sub "arn:aws:iam::${AWS::AccountId}:role/${ServiceRole}"
      DefaultInstanceProfileArn: !Sub "arn:aws:iam::${AWS::AccountId}:instance-profile/
${InstanceRole}"
      UseCustomCookbooks: 'true'
      CustomCookbooksSource:
        Type: git
        Url: git://github.com/amazonwebservices/opsworks-example-cookbooks.git
  myLayer:
    Type: AWS::OpsWorks::Layer
    DependsOn: myApp
    Properties:
      StackId:
        Ref: myStack
      Type: php-app
      Shortname: php-app
      EnableAutoHealing: 'true'
      AutoAssignElasticIps: 'false'
      AutoAssignPublicIps: 'true'
      Name: MyPHPApp
      CustomRecipes:
        Configure:
          - phpapp::appsetup
```

```
DBLayer:
  Type: AWS::OpsWorks::Layer
  DependsOn: myApp
  Properties:
    StackId:
      Ref: myStack
    Type: db-master
    Shortname: db-layer
    EnableAutoHealing: 'true'
    AutoAssignElasticIps: 'false'
    AutoAssignPublicIps: 'true'
    Name: MyMySQL
    CustomRecipes:
      Setup:
        - phpapp::dbsetup
    Attributes:
      MysqlRootPassword:
        Ref: MysqlRootPassword
      MysqlRootPasswordUbiquitous: 'true'
    VolumeConfigurations:
      - MountPoint: "/vol/mysql"
        NumberOfDisks: 1
        Size: 10
ELBAttachment:
  Type: AWS::OpsWorks::ElasticLoadBalancerAttachment
  Properties:
    ElasticLoadBalancerName:
      Ref: ELB
    LayerId:
      Ref: myLayer
ELB:
  Type: AWS::ElasticLoadBalancing::LoadBalancer
  Properties:
    AvailabilityZones:
      Fn::GetAZs: ''
    Listeners:
      - LoadBalancerPort: '80'
        InstancePort: '80'
        Protocol: HTTP
        InstanceProtocol: HTTP
    HealthCheck:
      Target: HTTP:80/
      HealthyThreshold: '2'
      UnhealthyThreshold: '10'
```

```
    Interval: '30'
    Timeout: '5'
myAppInstance1:
  Type: AWS::OpsWorks::Instance
  Properties:
    StackId:
      Ref: myStack
    LayerIds:
      - Ref: myLayer
    InstanceType: m1.small
myAppInstance2:
  Type: AWS::OpsWorks::Instance
  Properties:
    StackId:
      Ref: myStack
    LayerIds:
      - Ref: myLayer
    InstanceType: m1.small
myDBInstance:
  Type: AWS::OpsWorks::Instance
  Properties:
    StackId:
      Ref: myStack
    LayerIds:
      - Ref: DBLayer
    InstanceType: m1.small
myApp:
  Type: AWS::OpsWorks::App
  Properties:
    StackId:
      Ref: myStack
    Type: php
    Name:
      Ref: AppName
    AppSource:
      Type: git
      Url: git://github.com/amazonwebservices/opsworks-demo-php-simple-app.git
      Revision: version2
    Attributes:
      DocumentRoot: web
```

Cuplikan template Amazon Redshift

Amazon Redshift adalah layanan gudang data dengan skala petabyte yang terkelola penuh di cloud. Anda dapat menggunakan AWS CloudFormation untuk menyediakan dan mengelola kluster Amazon Redshift.

Kluster Amazon Redshift

Contoh template berikut membuat kluster Amazon Redshift sesuai dengan nilai parameter yang ditentukan saat tumpukan dibuat. Grup parameter cluster yang dikaitkan dengan cluster Amazon Redshift memungkinkan pencatatan aktivitas pengguna. Template ini juga meluncurkan cluster Amazon Redshift di VPC Amazon yang ditentukan dalam template. VPC menyertakan gateway internet sehingga Anda dapat mengakses kluster Amazon Redshift dari Internet. Namun, komunikasi antara kluster dan gateway Internet juga harus diaktifkan, yang dilakukan oleh entri tabel rute.

Note

Templat mencakup kondisi `IsMultiNodeCluster` sehingga parameter `NumberOfNodes` dinyatakan hanya ketika nilai parameter `ClusterType` diatur ke `multi-node`.

Contoh mendefinisikan parameter `MysqlRootPassword` dengan properti `NoEcho` diatur menjadi `true`. Jika Anda mengatur atribut `NoEcho` menjadi `true`, CloudFormation mengembalikan nilai parameter tertutup sebagai tanda bintang (`*****`) untuk setiap panggilan yang menggambarkan tumpukan atau peristiwa tumpukan, kecuali untuk informasi yang disimpan di lokasi yang ditentukan di bawah ini.

Important

Menggunakan `NoEcho` atribut tidak menutupi informasi apa pun yang disimpan sebagai berikut:

- Bagian Metadata template. CloudFormation tidak mengubah, memodifikasi, atau menyunting informasi apa pun yang Anda sertakan di Metadata bagian ini. Untuk informasi selengkapnya, lihat [Metadata](#).
- Bagian Outputs template. Untuk informasi selengkapnya, lihat [Output](#).
- MetadataAtribut definisi sumber daya. Untuk informasi lebih lanjut, lihat [Atribut Metadata](#).

Kami sangat menyarankan Anda untuk tidak menggunakan mekanisme ini untuk memasukkan informasi sensitif, seperti kata sandi atau rahasia.

Important

Daripada menyematkan informasi sensitif secara langsung di CloudFormation template Anda, kami sarankan Anda menggunakan parameter dinamis dalam template tumpukan untuk mereferensikan informasi sensitif yang disimpan dan dikelola di luar CloudFormation, seperti di AWS Systems Manager Parameter Store atau AWS Secrets Manager.

Untuk informasi selengkapnya, lihat [Do not embed credentials dalam praktik terbaik template Anda](#).

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

Node Type:

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
```


Lihat juga

[AWS::Redshift::Cluster](#)

Cuplikan template Amazon RDS

Topik

- [Sumber daya instans Amazon RDS DB](#)
- [Sumber daya instans DB database oracle Amazon RDS](#)
- [Sumber daya Amazon RDS DBSecurity Group untuk rentang CIDR](#)
- [Amazon RDS DBSecurity Group dengan grup EC2 keamanan Amazon](#)
- [Beberapa grup keamanan VPC](#)
- [Instans database Amazon RDS dalam grup keamanan VPC](#)

Sumber daya instans Amazon RDS DB

Contoh ini menunjukkan sumber daya Instans Amazon RDS DB dengan kata sandi pengguna master terkelola. Untuk informasi selengkapnya, lihat [Manajemen kata sandi dengan AWS Secrets Manager](#) di Panduan Pengguna Amazon RDS dan [Manajemen Kata AWS Secrets Manager Sandi](#) di Panduan Pengguna Aurora. Karena EngineVersion properti opsional tidak ditentukan, versi mesin default digunakan untuk Instans DB ini. Untuk detail tentang versi mesin default dan setelan default lainnya, lihat [Membuat DBInstance](#). DBSecurityGroupsProperti mengotorisasi masuknya jaringan ke AWS::RDS::DBSecurityGroup sumber daya bernama MyDbSecurityByEC2SecurityGroup dan MyDbSecurityByCIDRIPGroup Lihat rinciannya di [AWS::RDS::DBInstance](#). Sumber daya Instans DB juga memiliki DeletionPolicy atribut yang disetel keSnapshot. Dengan Snapshot DeletionPolicy set, AWS CloudFormation akan mengambil snapshot dari Instans DB ini sebelum menghapusnya selama penghapusan tumpukan.

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

```

Sumber daya instans DB database oracle Amazon RDS

Contoh ini membuat sumber daya Oracle Database DB Instance dengan kata sandi pengguna master dikelola. Untuk informasi selengkapnya, lihat [Manajemen kata sandi dengan AWS Secrets Manager](#) di Panduan Pengguna Amazon RDS. Contoh menentukan Engine as oracle-ee dengan model lisensi. bring-your-own-license [Untuk detail tentang pengaturan untuk instans Oracle Database DB, lihat Membuat. DBInstance](#) Properti DBSecurity Grup mengotorisasi masuknya jaringan ke AWS::RDS::DBSecurityGroup sumber daya bernama MyDbSecurityBy EC2 SecurityGroup dan. MyDbSecurityBy CIDRIPGroup Lihat rinciannya di [AWS::RDS::DBInstance](#). Sumber daya Instans DB juga memiliki DeletionPolicy atribut yang disetel keSnapshot. Dengan Snapshot DeletionPolicy set, AWS CloudFormation akan mengambil snapshot dari Instans DB ini sebelum menghapusnya selama penghapusan tumpukan.

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
```

Sumber daya Amazon RDS DBSecurity Group untuk rentang CIDR

Contoh ini menunjukkan DBSecurityGroup sumber daya Amazon RDS dengan otorisasi masuk untuk rentang CIDR yang ditentukan dalam format. ddd.ddd.ddd.ddd/dd [Untuk detailnya, lihat AWS: :RDS:: DBSecurity Group dan 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 RDS DBSecurity Group dengan grup EC2 keamanan Amazon

Contoh ini menunjukkan sumber daya [AWS: :RDS:: DBSecurity Grup](#) dengan otorisasi masuk dari grup EC2 keamanan Amazon yang direferensikan oleh. MyEc2SecurityGroup

Untuk melakukan ini, Anda mendefinisikan grup EC2 keamanan dan kemudian menggunakan Ref fungsi intrinsik untuk merujuk ke grup EC2 keamanan di dalam Anda. 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

Contoh ini diekstrak dari contoh lengkap berikut: [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

```

Beberapa grup keamanan VPC

Contoh ini menunjukkan [AWS::RDS::DBSecurity Group](#) resource dengan otorisasi ingress untuk beberapa grup keamanan Amazon EC2 VPC di [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'

```

Instans database Amazon RDS dalam grup keamanan VPC

Contoh ini menunjukkan instance database Amazon RDS yang terkait dengan grup keamanan Amazon EC2 VPC.

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

```

Cuplikan templat Route 53

Topik

- [Rekaman sumber daya Amazon Route 53 yang ditetapkan menggunakan nama atau ID zona yang dihosting](#)
- [Menggunakan RecordSetGroup untuk mengatur set catatan sumber daya tertimbang](#)
- [Menggunakan RecordSetGroup untuk menyiapkan kumpulan catatan sumber daya alias](#)
- [Catatan sumber daya alias ditetapkan untuk distribusi CloudFront](#)

Rekaman sumber daya Amazon Route 53 yang ditetapkan menggunakan nama atau ID zona yang dihosting

Saat Anda membuat kumpulan catatan sumber daya Amazon Route 53, Anda harus menentukan zona yang dihosting tempat Anda ingin menambahkannya. CloudFormation menyediakan dua cara untuk menentukan zona yang dihosting:

- Anda dapat secara eksplisit menentukan zona yang di-host menggunakan properti `HostedZoneId`.
- Anda dapat CloudFormation menemukan zona yang dihosting menggunakan `HostedZoneName` properti. Jika Anda menggunakan `HostedZoneName` properti dan ada beberapa zona yang dihosting dengan nama yang sama, CloudFormation tidak membuat tumpukan.

Menambahkan RecordSet menggunakan HostedZoneId

Contoh ini menambahkan kumpulan data sumber daya Amazon Route 53 yang berisi SPF catatan untuk nama domain `mysite.example.com` yang menggunakan `HostedZoneId` properti untuk menentukan zona yang dihosting.

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"'
```

Menambahkan RecordSet menggunakan HostedZoneName

Contoh ini menambahkan data sumber daya Amazon Route 53 yang ditetapkan untuk nama domain “mysite.example.com” menggunakan HostedZoneName properti untuk menentukan zona yang dihosting.

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
```

Menggunakan RecordSetGroup untuk mengatur set catatan sumber daya tertimbang

Contoh ini menggunakan [AWS::Route53::RecordSetGroup](#) untuk menyiapkan dua catatan CNAME untuk "example.com." zona yang di-host. Properti RecordSets mengandung set catatan CNAME untuk nama DNS "mysite.example.com". Setiap set catatan mengandung sebuah pengidentifikasi (SetIdentifier) dan bobot (Weight). Proporsi lalu lintas internet yang dirutekan ke sumber daya didasarkan pada perhitungan berikut:

- Frontend One: $140 / (140 + 60) = 140 / 200 = 70\%$
- Frontend Two: $60 / (140 + 60) = 60 / 200 = 30\%$

Untuk informasi selengkapnya tentang kumpulan rekaman sumber daya tertimbang, lihat [Perutean tertimbang di Panduan Pengembang](#) 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

```

Menggunakan RecordSetGroup untuk menyiapkan kumpulan catatan sumber daya alias

Contoh berikut menggunakan [AWS::Route53::RecordSetGroup](#) untuk menyiapkan kumpulan catatan sumber daya alias bernama `example.com` yang merutekan lalu lintas ke penyeimbang beban ELB Versi 1 (Klasik) dan penyeimbang beban Versi 2 (Aplikasi atau Jaringan). [AliasTarget](#) Properti menentukan ID zona host dan nama DNS untuk myELB LoadBalancer dengan menggunakan fungsi `GetAtt` intrinsik. `GetAtt` mengambil properti myELB sumber daya yang berbeda, tergantung pada apakah Anda merutekan lalu lintas ke penyeimbang beban Versi 1 atau Versi 2:

- Penyeimbang beban Versi 1: `CanonicalHostedZoneNameID` dan `DNSName`
- Penyeimbang beban Versi 2: `CanonicalHostedZoneID` dan `DNSName`

Untuk informasi selengkapnya tentang kumpulan rekaman sumber daya alias, lihat [Memilih antara catatan alias dan non-alias](#) di Panduan Pengembang Route 53.

JSON untuk penyeimbang beban versi 1

```

"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 untuk penyeimbang beban versi 1

```

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 untuk penyeimbang beban versi 2

```

"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 untuk penyeimbang beban versi 2

```
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'
```

Catatan sumber daya alias ditetapkan untuk distribusi CloudFront

Contoh berikut membuat catatan alias A yang mengarahkan nama domain kustom ke CloudFront distribusi yang ada. myHostedZoneID diasumsikan sebagai referensi ke AWS::Route53::HostedZone sumber daya aktual dalam template atau parameter yang sama. myCloudFrontDistribution mengacu pada sumber AWS::CloudFront::Distribution daya dalam template yang sama. Catatan alias menggunakan ID zona CloudFront host standar (Z2FDTNDATAQYW2) dan secara otomatis menyelesaikan nama domain distribusi menggunakan Fn::GetAtt. Pengaturan ini memungkinkan lalu lintas web dialihkan dari domain khusus ke CloudFront distribusi tanpa memerlukan alamat IP.

Note

Ketika Anda membuat kumpulan catatan sumber daya alias, Anda harus menentukan Z2FDTNDATAQYW2 untuk HostedZoneId properti. Kumpulan catatan sumber daya alias untuk tidak CloudFront dapat dibuat di zona pribadi.

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
```

Cuplikan template Amazon S3

Gunakan contoh template Amazon S3 ini untuk membantu mendeskripsikan bucket Amazon S3 Anda. CloudFormation Untuk contoh lainnya, lihat bagian [Contoh](#) di `AWS::S3::Bucket` sumber daya.

Topik

- [Membuat bucket Amazon S3 dengan default](#)
- [Membuat bucket Amazon S3 untuk hosting situs web dan dengan DeletionPolicy](#)
- [Pembuatan situs web statis menggunakan domain kustom](#)

Membuat bucket Amazon S3 dengan default

Contoh ini menggunakan a [AWS::S3::Bucket](#) untuk membuat bucket dengan pengaturan default.

JSON

```
"myS3Bucket" : {
  "Type" : "AWS::S3::Bucket"
}
```

YAML

```
MyS3Bucket:
  Type: AWS::S3::Bucket
```

Membuat bucket Amazon S3 untuk hosting situs web dan dengan **DeletionPolicy**

Contoh ini membuat bucket sebagai situs web dan menonaktifkan Blokir Akses Publik (izin baca publik diperlukan untuk bucket yang disiapkan untuk hosting situs web). Kebijakan bucket publik kemudian ditambahkan ke bucket. Karena sumber daya bucket ini memiliki `DeletionPolicy` atribut yang disetel ke `Retain`, tidak CloudFormation akan menghapus bucket ini saat menghapus tumpukan. OutputBagian ini digunakan `Fn::GetAtt` untuk mengambil `WebsiteURL` atribut dan `DomainName` atribut `S3Bucket` sumber daya.

Note

Contoh berikut mengasumsikan pengaturan BlockPublicPolicy dan RestrictPublicBuckets Blokir Akses Publik telah dinonaktifkan di tingkat akun.

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

```

Pembuatan situs web statis menggunakan domain kustom

Anda dapat menggunakan Route 53 dengan domain terdaftar. Contoh berikut mengasumsikan bahwa Anda telah membuat zona yang dihosting di Route 53 untuk domain Anda. Contoh ini membuat dua bucket untuk hosting situs web. Bucket root meng-host konten, dan bucket lainnya mengalihkan permintaan `www.domainname.com` ke bucket root. Kumpulan catatan memetakan nama domain Anda ke titik akhir Amazon S3.

Anda juga perlu menambahkan kebijakan bucket, seperti yang ditunjukkan pada contoh di atas.

Untuk informasi selengkapnya tentang penggunaan domain kustom, lihat [Tutorial: Mengonfigurasi situs web statis menggunakan domain kustom yang terdaftar dengan Route 53](#) di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

Note

Contoh berikut mengasumsikan pengaturan `BlockPublicPolicy` dan `RestrictPublicBuckets` Blokir Akses Publik telah dinonaktifkan di tingkat akun.

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

```

Cuplikan template Amazon SNS

Contoh ini menunjukkan sumber daya topik Amazon SNS. Ini membutuhkan alamat email yang valid.

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
```

Cuplikan template Amazon SQS

Contoh ini menunjukkan antrian Amazon SQS.

JSON

```
"MyQueue" : {
  "Type" : "AWS::SQS::Queue",
  "Properties" : {
    "VisibilityTimeout" : "value"
  }
}
```

YAML

```
MyQueue:
  Type: AWS::SQS::Queue
  Properties:
    VisibilityTimeout: value
```

Cuplikan template Amazon Timestream

Amazon TimeStream untuk InfluxDB memudahkan pengembang dan DevOps tim aplikasi untuk menjalankan database InfluxDB yang dikelola sepenuhnya untuk aplikasi deret waktu nyata menggunakan sumber terbuka. AWS APIs Anda dapat dengan cepat membuat database InfluxDB yang menangani beban kerja deret waktu yang menuntut. Dengan beberapa panggilan API sederhana, Anda dapat mengatur, memigrasi, mengoperasikan, dan menskalakan database InfluxDB AWS dengan penambalan, pencadangan, dan pemulihan perangkat lunak otomatis. Anda juga dapat menemukan sampel ini di [awslabs/amazon-timestream-tools/tree/mainline/integrations/cloudformation/timestream-influxdb](https://github.com/aws-labs/amazon-timestream-tools/tree/mainline/integrations/cloudformation/timestream-influxdb) on. GitHub

Topik

- [Sampel minimal menggunakan nilai default](#)
- [Contoh yang lebih lengkap dengan parameter](#)

AWS CloudFormation Template ini membuat sumber daya berikut yang diperlukan untuk berhasil membuat, menyambungkan, dan memantau Amazon Timestream untuk instans InfluxDB:

Amazon VPC

- VPC
- Satu atau lebih Subnet
- InternetGateway
- RouteTable
- SecurityGroup

Amazon S3

- Bucket

Amazon Timestream

- InfluxDBInstance

Sampel minimal menggunakan nilai default

Contoh ini menyebarkan instans multi-AZ dan dapat diakses publik menggunakan nilai default jika memungkinkan.

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::VPCElasticNetworkInterfaceAttachment

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"]]
```

Contoh yang lebih lengkap dengan parameter

Contoh template ini secara dinamis mengubah sumber daya jaringan berdasarkan parameter yang disediakan. Parameter termasuk `PubliclyAccessible` dan `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",

```

```

        "InfluxI0IncludedT3"
    ]
},
"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"] ]
```


Menyebarkan tumpukan Windows berbasis menggunakan CloudFormation

Halaman ini menyediakan tautan ke dokumentasi referensi teknis untuk CloudFormation sumber daya yang biasa digunakan dalam penerapan Windows berbasis.

CloudFormation memberikan dukungan untuk menyebarkan dan mengelola Microsoft Windows tumpukan melalui. Infrastructure as Code (IaC) Anda dapat menggunakan CloudFormation untuk penyediaan otomatis EC2 instans Windows berbasis, di SQL Server Amazon RDS, dan melalui. Microsoft Active Directory AWS Directory Service

AWS menyediakan Amazon Machine Images (AMIs) yang telah dikonfigurasi sebelumnya yang dirancang khusus untuk Windows platform untuk membantu Anda menerapkan aplikasi dengan cepat di Amazon. EC2 Ini AMIs termasuk Microsoft pengaturan default dan penyesuaian AWS-spesifik. Dengan CloudFormation, Anda dapat memilih AMI yang sesuai, meluncurkan instance, dan mengaksesnya menggunakan Remote Desktop Connection, seperti yang Anda lakukan dengan yang lain Windows Server. Ini AMIs berisi komponen perangkat lunak penting, termasuk EC2Launch (versi bervariasi menurut Windows Server edisi),, AWS Systems Manager, CloudFormation, Alat AWS untuk PowerShell, dan berbagai driver jaringan, penyimpanan, dan grafis untuk memastikan kinerja dan kompatibilitas yang optimal dengan AWS layanan. Untuk informasi selengkapnya, lihat [Referensi AWS Windows AMI](#).

CloudFormation juga mendukung alat konfigurasi perangkat lunak, seperti UserData skrip, yang dapat menjalankan PowerShell atau perintah batch ketika EC2 instance pertama kali boot up. Ini juga menawarkan skrip pembantu (cfn-init,, cfn-signalcfn-get-metadata, dan cfn-hup) dan mendukung `AWS::CloudFormation::Init` metadata untuk mengelola paket, file, dan layanan pada instance. Windows

Untuk lingkungan perusahaan, CloudFormation memungkinkan penggabungan domain, manajemen Windows lisensi melalui model EC2 lisensi, dan penanganan kredensi yang aman dengan. AWS Secrets Manager Dikombinasikan dengan template yang dikontrol versi dan penerapan berulang, CloudFormation membantu organisasi mempertahankan lingkungan yang konsisten, aman, dan terukur di beberapa akun dan akun Windows. Wilayah AWS

Untuk detail tentang CloudFormation sumber daya yang biasa digunakan dalam penerapan Windows berbasis, lihat topik referensi teknis berikut.

Jenis sumber daya	Deskripsi
AWS::EC2::Instance	Untuk meluncurkan Windows EC2 instance.
AWS::EC2::SecurityGroup	Untuk menentukan aturan firewall untuk Windows beban kerja.
AWS::AutoScaling::AutoScalingGroup	Untuk Windows EC2 instance penskalaan.
AWS::EC2::LaunchTemplate	
AWS::DirectoryService::MicrosoftKLAN	Untuk menyebarkan MicrosoftActive Directory.
AWS::FSx::FileSystem	Untuk menyebarkan FSx untuk Windows File Server.
AWS::RDS::DBInstance	Untuk penyediaan di SQL Server Amazon RDS.
AWS::CloudFormation::Init	Digunakan dalam EC2 metadata untuk mengkonfigurasi instance. Untuk informasi selengkapnya, lihat Tumpukan berbasis bootstrap Windows CloudFormation .
AWS::SecretsManager::Secret	Untuk mengelola kredensi dan kata sandi dengan aman. Windows
AWS::SSM::Parameter	Untuk menyimpan nilai konfigurasi dengan aman.
AWS::IAM::InstanceProfile AWS::IAM::Role	Untuk memberikan izin ke aplikasi yang berjalan pada EC2 instance.

Tumpukan berbasis bootstrap Windows CloudFormation

Topik ini menjelaskan cara mem-bootstrap Windows tumpukan dan memecahkan masalah pembuatan tumpukan.

Topik

- [Data pengguna dalam EC2 contoh](#)
- [CloudFormation skrip pembantu](#)
- [Contoh bootstrap tumpukan Windows](#)
- [Kelola Windows layanan](#)
- [Memecahkan masalah pembuatan tumpukan](#)

Data pengguna dalam EC2 contoh

Data pengguna adalah EC2 fitur Amazon yang memungkinkan Anda meneruskan skrip atau informasi konfigurasi ke EC2 instance saat diluncurkan.

Untuk Windows EC2 contoh:

- Data pengguna dijalankan sebagai skrip batch.
- Anda dapat menggunakan perintah batch (menggunakan `<script>` tag) atau PowerShell skrip (menggunakan `<powershell>` tag).
- Eksekusi skrip ditangani oleh EC2Launch.

Important

Jika Anda membuat Windows AMI sendiri untuk digunakan CloudFormation, pastikan EC2Launch v2 dikonfigurasi dengan benar. EC2Launchv2 diperlukan untuk alat CloudFormation bootstrap untuk menginisialisasi dan mengonfigurasi Windows instance dengan benar selama pembuatan tumpukan. Untuk informasi selengkapnya, lihat [Menggunakan agen EC2Launch v2 untuk melakukan tugas selama peluncuran instans EC2 Windows](#) di Panduan EC2 Pengguna Amazon.

Untuk selengkapnya AWS Windows AMIs, lihat [Referensi AWSWindows AMI](#).

CloudFormation skrip pembantu

Skrip pembantu adalah utilitas untuk mengkonfigurasi instance selama proses bootstrap. Digunakan dengan data EC2 pengguna Amazon, mereka menyediakan opsi konfigurasi yang kuat.

CloudFormation menyediakan skrip pembantu Python berikut yang dapat Anda gunakan untuk menginstal perangkat lunak dan memulai layanan pada EC2 instance Amazon yang Anda buat sebagai bagian dari tumpukan Anda:

- `cfn-init`— Gunakan untuk mengambil dan menafsirkan metadata sumber daya, menginstal paket, membuat file, dan memulai layanan.
- `cfn-signal`— Gunakan untuk memberi sinyal dengan `CreationPolicy` atau `WaitCondition`, sehingga Anda dapat menyinkronkan sumber daya lain di tumpukan ketika sumber daya atau aplikasi prasyarat siap.
- `cfn-get-metadata`— Gunakan untuk mengambil metadata untuk sumber daya atau jalur ke kunci tertentu.
- `cfn-hup`— Gunakan untuk memeriksa pembaruan metadata dan menjalankan kait khusus saat perubahan terdeteksi.

Anda memanggil skrip langsung dari templat Anda. Skrip bekerja bersama dengan metadata sumber daya yang didefinisikan dalam templat yang sama. Skrip berjalan di EC2 instance Amazon selama proses pembuatan tumpukan.

Untuk informasi selengkapnya, lihat [referensi skrip CloudFormation pembantu di Panduan Referensi AWS CloudFormation](#) Template.

Contoh bootstrap tumpukan Windows

Mari kita periksa contoh cuplikan dari template SharePoint server yang melakukan tindakan berikut:

- Mengkonfigurasi file inialisasi: `cfn-credentials`, `cfn-hup.conf`, dan `cfn-auto-reloader.conf`
- Mengunduh dan menginstal paket seperti SharePoint Foundation pada instance server.
- Menggunakan a `WaitCondition` untuk memastikan sumber daya siap.
- Membuat pengguna IAM dan grup keamanan untuk akses ke instance.
- Mengambil IP untuk instance dengan Amazon Elastic IP (EIP).

Skrip `cfn-init` pembantu digunakan untuk melakukan setiap tindakan ini, berdasarkan informasi dalam `AWS::CloudFormation::Init` sumber daya dalam template.

`AWS::CloudFormation::Init` Bagian ini diberi nama `SharePointFoundation` dan dimulai dengan deklarasi standar:

```
"SharePointFoundation": {
```

```
"Type" : "AWS::EC2::Instance",
"Metadata" : {
  "AWS::CloudFormation::Init" : {
    "config" : {
```

Setelah ini, files bagian dari `AWS::CloudFormation::Init` dinyatakan:

```
"files" : {
  "c:\\cfn\\cfn-hup.conf" : {
    "content" : { "Fn::Join" : [ "", [
      "[main]\\n",
      "stack=", { "Ref" : "AWS::StackName" }, "\\n",
      "region=", { "Ref" : "AWS::Region" }, "\\n"
    ] ] }
  },
  "c:\\cfn\\hooks.d\\cfn-auto-reloader.conf" : {
    "content": { "Fn::Join" : [ "", [
      "[cfn-auto-reloader-hook]\\n",
      "triggers=post.update\\n",
      "path=Resources.SharePointFoundation.Metadata.AWS::CloudFormation::Init\\n",
      "action=cfn-init.exe -v -s ", { "Ref" : "AWS::StackName" },
      " -r SharePointFoundation",
      " --region ", { "Ref" : "AWS::Region" }, "\\n"
    ] ] }
  },
  "C:\\\\SharePoint\\\\SharePointFoundation2010.exe" : {
    "source" : "http://d3adzpja92utk0.cloudfront.net/SharePointFoundation.exe"
  }
},
```

Tiga file dibuat di sini dan ditempatkan di direktori `C:\cfn` pada instans server. Mereka adalah:

- `cfn-hup.conf`, file konfigurasi untuk `cfn-hup`.
 - `cfn-auto-reloader.conf`, file konfigurasi untuk hook yang digunakan oleh `cfn-hup` untuk memulai pembaruan (panggilan `cfn-init`) saat metadata berubah.
- `AWS::CloudFormation::Init`

Ada juga file yang diunduh ke server: `SharePointFoundation.exe`. File ini digunakan untuk menginstal SharePoint pada instance server.

⚠ Important

Karena jalur Windows menggunakan karakter garis miring terbalik ('\\'), Anda harus selalu ingat untuk menghindari semua garis miring terbalik dengan benar dengan mendahului garis miring terbalik lainnya setiap kali Anda merujuk ke jalur di template. Windows CloudFormation

Berikutnya adalah commands bagian, yang merupakan cmd.exe perintah.

```
"commands" : {
  "1-extract" : {
    "command" : "C:\\\\SharePoint\\\\SharePointFoundation2010.exe /extract:C:\\\\SharePoint\\
\\SPF2010 /quiet /log:C:\\\\SharePoint\\\\SharePointFoundation2010-extract.log"
  },
  "2-prereq" : {
    "command" : "C:\\\\SharePoint\\\\SPF2010\\\\PrerequisiteInstaller.exe /unattended"
  },
  "3-install" : {
    "command" : "C:\\\\SharePoint\\\\SPF2010\\\\setup.exe /config C:\\\\SharePoint\\\\SPF2010\\
\\Files\\\\SetupSilent\\\\config.xml"
  }
}
```

Karena perintah dalam instans diproses dalam urutan abjad berdasarkan nama, setiap perintah telah ditambahkan dengan nomor yang menunjukkan urutan eksekusi yang diinginkan. Dengan demikian, kita dapat memastikan bahwa paket instalasi pertama diekstraksi, semua prasyarat kemudian diinstal, dan akhirnya, instalasi dimulai. SharePoint

Berikutnya adalah Properties bagian:

```
"Properties": {
  "InstanceType" : { "Ref" : "InstanceType" },
  "ImageId" : { "Fn::FindInMap" : [ "AWSRegionArch2AMI", { "Ref" : "AWS::Region" },
    { "Fn::FindInMap" : [ "AWSInstanceType2Arch", { "Ref" :
"InstanceType" }, "Arch" ] } ] ],
  "SecurityGroups" : [ { "Ref" : "SharePointFoundationSecurityGroup" } ],
  "KeyName" : { "Ref" : "KeyPairName" },
  "UserData" : { "Fn::Base64" : { "Fn::Join" : [ "", [
    "<script>\n",

    "cfn-init.exe -v -s ", { "Ref" : "AWS::StackName" },
    " -r SharePointFoundation",
```

```

    " --region ", { "Ref" : "AWS::Region" }, "\n",

    "cfn-signal.exe -e %ERRORLEVEL% ", { "Fn::Base64" : { "Ref" :
"SharePointFoundationWaitHandle" }}, "\n",

    "</script>"
  ]]]}
}

```

Di bagian ini, UserData properti berisi cmd.exe skrip yang akan dieksekusi oleh cfn-init, dikelilingi oleh <script> tag. Anda dapat menggunakan Windows PowerShell skrip di sini sebagai gantinya dengan mengelilingi skrip Anda dengan <powershell> tag. Untuk Windows tumpukan, Anda harus base64 menyandikan URL penanganan kondisi tunggu lagi.

SharePointFoundationWaitHandle direferensikan di sini dan dijalankan dengan cfn-signal. Yang WaitConditionHandle dan terkait WaitCondition dideklarasikan berikutnya dalam template:

```

"SharePointFoundationWaitHandle" : {
  "Type" : "AWS::CloudFormation::WaitConditionHandle"
},

"SharePointFoundationWaitCondition" : {
  "Type" : "AWS::CloudFormation::WaitCondition",
  "DependsOn" : "SharePointFoundation",
  "Properties" : {
    "Handle" : {"Ref" : "SharePointFoundationWaitHandle"},
    "Timeout" : "3600"
  }
}

```

Karena menjalankan semua langkah dan instalasi SharePoint mungkin memakan waktu cukup lama, tetapi tidak satu jam penuh, WaitCondition menunggu satu jam (3600 detik) sebelum waktu habis.

Jika semuanya berjalan dengan baik, IP Elastis digunakan untuk menyediakan akses ke SharePoint instance:

```

"Outputs" : {
  "SharePointFoundationURL" : {
    "Value" : { "Fn::Join" : [ "", [ "http://", { "Ref" :
"SharePointFoundationEIP" } ] ] },

```

```
"Description" : "SharePoint Team Site URL. Please retrieve Administrator password of the instance and use it to access the URL"
}
```

Setelah pembuatan tumpukan selesai, alamat IP yang disediakan oleh EIP akan ditampilkan di tab Output konsol. CloudFormation Namun, sebelum Anda dapat mengakses instance, Anda perlu mengambil kata sandi administrator sementara yang dihasilkan untuk instance tersebut. Untuk informasi selengkapnya, lihat [Connect ke Windows instans menggunakan RDP](#) di EC2 Panduan Pengguna Amazon.

Kelola Windows layanan

Anda mengelola Windows layanan dengan cara yang sama seperti layanan Linux, kecuali bahwa Anda menggunakan windows kunci, bukan sysvinit. Contoh berikut memulai cfn-hup layanan, menyetelnya ke Otomatis, dan memulai ulang layanan jika cfn-init memodifikasi c:\cfn\cfn-hup.conf atau file c:\cfn\hooks.d\cfn-auto-reloader.conf konfigurasi.

```
"services" : {
  "windows" : {
    "cfn-hup" : {
      "enabled" : "true",
      "ensureRunning" : "true",
      "files" : ["c:\\cfn\\cfn-hup.conf", "c:\\cfn\\hooks.d\\cfn-auto-reloader.conf"]
    }
  }
}
```

Anda dapat mengelola Windows layanan lain dengan cara yang sama dengan menggunakan nama—bukan nama tampilan—untuk mereferensikan layanan.

Memecahkan masalah pembuatan tumpukan

Jika tumpukan Anda gagal selama pembuatan, perilaku defaultnya adalah mengembalikan kegagalan. Meskipun ini biasanya merupakan default yang baik karena menghindari biaya yang tidak perlu, ini menyulitkan untuk men-debug mengapa pembuatan tumpukan Anda gagal.

Untuk menonaktifkan perilaku ini saat membuat atau memperbarui tumpukan Anda dengan CloudFormation konsol, pilih opsi Pertahankan sumber daya yang berhasil disediakan di bawah opsi Kegagalan tumpukan. Untuk informasi selengkapnya, lihat [Pilih cara menangani kegagalan saat menyediakan sumber daya](#). Ini memungkinkan Anda untuk masuk ke instance Anda dan melihat file log untuk menentukan masalah yang dihadapi saat menjalankan skrip startup Anda.

Log penting yang perlu dilihat adalah:

- Log EC2 konfigurasi di %ProgramData%\Amazon\EC2Launch\log\agent.log
- Log cfn-init pada C:\cfn\log\cfn-init.log

Untuk log lainnya, lihat topik berikut di Panduan EC2 Pengguna Amazon:

- [EC2Launchstruktur direktori](#)
- [EC2Launchstruktur direktori v2](#)

Perluas kemampuan template Anda dengan CloudFormation tipe sumber daya yang disediakan

CloudFormation menawarkan beberapa jenis sumber daya yang dapat Anda gunakan dalam template tumpukan Anda untuk memperluas kemampuannya di luar template tumpukan sederhana.

Jenis sumber daya ini meliputi:

Jenis sumber daya	Deskripsi	Dokumentasi
Sumber daya khusus	Jenis <code>AWS::CloudFormation::CustomResource</code> sumber daya memungkinkan Anda membuat sumber daya khusus yang dapat melakukan tugas penyediaan tertentu atau menyertakan sumber daya yang tidak tersedia sebagai jenis CloudFormation sumber daya.	Sumber daya khusus
Makro	Jenis <code>AWS::CloudFormation::Macro</code> sumber daya mendefinisikan potongan kode yang dapat digunakan kembali yang dapat melakukan pemrosesan khusus pada CloudFormation templat. Makro dapat memodifikasi template Anda, menghasilkan sumber daya tambahan, atau melakukan operasi kustom lainnya selama pembuatan tumpukan atau pembaruan.	Makro templat

Jenis sumber daya	Deskripsi	Dokumentasi
Tumpukan nested	Jenis <code>AWS::CloudFormation::Stack</code> sumber daya memungkinkan Anda membuat tumpukan bersarang di dalam CloudFormation templat Anda untuk arsitektur tumpukan yang lebih modular dan dapat digunakan kembali.	Tumpukan nested
StackSet	Jenis <code>AWS::CloudFormation::StackSet</code> sumber daya membuat atau memperbarui a CloudFormation StackSet, yang merupakan wadah untuk tumpukan yang dapat digunakan di beberapa Akun AWS dan Wilayah.	Mengelola tumpukan dengan StackSets
Kondisi menunggu	Jenis <code>AWS::CloudFormation::WaitCondition</code> sumber daya menghentikan pembuatan tumpukan atau pembaruan hingga kondisi tertentu terpenuhi, seperti keberhasilan penyelesaian proses yang berjalan lama atau ketersediaan sumber daya eksternal.	Kondisi tunggu
Tunggu kondisi pegangan	Jenis <code>AWS::CloudFormation::WaitConditionHandle</code> sumber daya bekerja sama dengan tipe <code>AWS::CloudFormation::WaitCondition</code> sumber daya. Ini menyediakan URL presigned yang digunakan untuk mengirim sinyal yang menunjukkan bahwa kondisi tertentu telah terpenuhi. Sinyal-sinyal ini memungkinkan pembuatan tumpukan atau proses pembaruan untuk dilanjutkan.	Kondisi tunggu

Buat logika penyediaan khusus dengan sumber daya khusus

Sumber daya khusus menyediakan cara bagi Anda untuk menulis logika penyediaan khusus ke dalam CloudFormation templat Anda dan CloudFormation menjalankannya kapan saja Anda membuat, memperbarui (jika Anda mengubah sumber daya khusus), atau menghapus tumpukan. Ini

dapat berguna ketika persyaratan penyediaan Anda melibatkan logika kompleks atau alur kerja yang tidak dapat diekspresikan dengan CloudFormation tipe sumber daya bawaan.

Misalnya, Anda mungkin ingin menyertakan sumber daya yang tidak tersedia sebagai tipe CloudFormation sumber daya. Anda dapat menyertakan sumber daya tersebut dengan menggunakan sumber daya kustom. Dengan begitu, Anda masih dapat mengelola semua sumber daya terkait Anda dalam satu tumpukan.

Untuk menentukan sumber daya kustom dalam CloudFormation template Anda, Anda menggunakan `AWS::CloudFormation::CustomResource` atau jenis `Custom::MyCustomResourceTypeName` sumber daya. Sumber daya khusus memerlukan satu properti, token layanan, yang menentukan ke mana permintaan CloudFormation dikirim, seperti topik Amazon SNS atau fungsi Lambda.

Topik berikut memberikan informasi tentang cara menggunakan sumber daya khusus.

Topik

- [Cara kerja sumber daya kustom](#)
- [Batas waktu respons](#)
- [Sumber daya khusus yang didukung Amazon SNS](#)
- [Sumber daya khusus yang didukung Lambda](#)
- [Referensi sumber daya kustom](#)

Note

CloudFormation Registri dan sumber daya khusus masing-masing menawarkan manfaatnya sendiri. Sumber daya khusus menawarkan manfaat berikut:

- Anda tidak perlu mendaftarkan sumber daya.
- Anda dapat menyertakan seluruh sumber daya sebagai bagian dari templat tanpa mendaftar.
- Mendukung `Create`, `Update`, dan `Delete` operasi

Keuntungan yang ditawarkan sumber daya berbasis registri meliputi:

- Mendukung pemodelan, penyediaan, dan pengelolaan sumber daya aplikasi pihak ketiga
- Mendukung operasi `Create`, `Read`, `Update`, `Delete`, dan `List` (CRUDL)

- Mendukung deteksi penyimpangan pada jenis sumber daya privat dan pihak ketiga

Tidak seperti sumber daya khusus, sumber daya berbasis registri tidak perlu mengaitkan topik Amazon SNS atau fungsi Lambda untuk melakukan operasi. CRUDL Untuk informasi selengkapnya, lihat [Mengelola ekstensi dengan CloudFormation registri](#).

Cara kerja sumber daya kustom

Proses umum untuk menyiapkan sumber daya kustom baru mencakup langkah-langkah berikut. Langkah-langkah ini melibatkan dua peran: penyedia sumber daya kustom yang memiliki sumber daya kustom dan pengembang template yang membuat template yang menyertakan jenis sumber daya kustom. Ini bisa menjadi orang yang sama, tetapi jika tidak, penyedia sumber daya khusus harus bekerja dengan pengembang template.

1. Penyedia sumber daya kustom menulis logika yang menentukan cara menangani permintaan dari CloudFormation dan melakukan tindakan pada sumber daya kustom.
2. Penyedia sumber daya khusus membuat topik Amazon SNS atau fungsi Lambda tempat CloudFormation dapat mengirim permintaan. Topik Amazon SNS atau fungsi Lambda harus berada di Wilayah yang sama tempat tumpukan akan dibuat.
3. Penyedia sumber daya khusus memberikan ARN topik Amazon SNS atau ARN fungsi Lambda kepada pengembang template.
4. Pengembang template mendefinisikan sumber daya kustom dalam CloudFormation template mereka. Ini termasuk token layanan dan parameter data input apa pun. Token layanan dan struktur data input ditentukan oleh penyedia sumber daya khusus. Token layanan menentukan topik Amazon SNS ARN atau fungsi Lambda ARN dan selalu diperlukan, tetapi data input bersifat opsional tergantung pada sumber daya khusus.

Sekarang, setiap kali ada yang menggunakan template untuk membuat, memperbarui, atau menghapus sumber daya kustom, CloudFormation mengirimkan permintaan ke token layanan yang ditentukan, dan kemudian menunggu respons sebelum melanjutkan dengan operasi tumpukan.

Berikut ini merangkum alur untuk membuat tumpukan dari template:

1. CloudFormation mengirimkan permintaan ke token layanan yang ditentukan. Permintaan tersebut mencakup informasi seperti jenis permintaan dan URL Layanan Penyimpanan Amazon Simple yang telah ditandatangani sebelumnya, tempat sumber daya kustom mengirimkan tanggapan.

Untuk informasi selengkapnya tentang apa saja yang disertakan dalam permintaan, lihat [Objek permintaan sumber daya kustom](#).

Data sampel berikut menunjukkan apa yang CloudFormation termasuk dalam Create permintaan. Dalam contoh ini, ResourceProperties memungkinkan CloudFormation untuk membuat payload khusus untuk dikirim ke fungsi Lambda.

```
{
  "RequestType" : "Create",
  "ResponseURL" : "http://pre-signed-S3-url-for-response",
  "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-cd98-11ea-90d5-0a9cd3354c10",
  "RequestId" : "unique id for this create request",
  "ResourceType" : "Custom::TestResource",
  "LogicalResourceId" : "MyTestResource",
  "ResourceProperties" : {
    "Name" : "Value",
    "List" : [ "1", "2", "3" ]
  }
}
```

2. Penyedia sumber daya kustom memproses CloudFormation permintaan dan mengembalikan respons SUCCESS atau FAILED ke URL yang telah ditandatangani sebelumnya. Penyedia sumber daya khusus memberikan respons dalam file berformat JSON dan mengunggahnya ke URL S3 yang telah ditandatangani sebelumnya. Untuk informasi selengkapnya, lihat [Mengunggah objek dengan presigned URLs](#) di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

Sebagai tanggapan, penyedia sumber daya kustom juga dapat menyertakan pasangan nama-nilai yang dapat diakses oleh pengembang template. Misalnya, respons dapat menyertakan data output jika permintaan berhasil atau pesan kesalahan jika permintaan gagal. Untuk informasi selengkapnya tentang respons, lihat [Objek tanggapan sumber daya khusus](#).

Important

Jika pasangan nama-nilai berisi informasi sensitif, Anda harus menggunakan NoEcho bidang untuk menutupi output dari sumber daya kustom. Jika tidak, nilai terlihat melalui APIs nilai properti permukaan (seperti DescribeStackEvents).

Untuk informasi selengkapnya tentang penggunaan NoEcho untuk menutupi informasi sensitif, lihat praktik [Jangan menanamkan kredensial dalam templat Anda](#) terbaik.

Penyedia sumber daya khusus bertanggung jawab untuk mendengarkan dan menanggapi permintaan. Misalnya, untuk notifikasi Amazon SNS, penyedia sumber daya khusus harus mendengarkan dan menanggapi notifikasi yang dikirim ke topik tertentu ARN. CloudFormation menunggu dan mendengarkan respons di lokasi URL yang telah ditandatangani sebelumnya.

Data contoh berikut menunjukkan apa yang mungkin disertakan oleh sumber daya kustom dalam respons:

```
{
  "Status" : "SUCCESS",
  "PhysicalResourceId" : "TestResource1",
  "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-
cd98-11ea-90d5-0a9cd3354c10",
  "RequestId" : "unique id for this create request",
  "LogicalResourceId" : "MyTestResource",
  "Data" : {
    "OutputName1" : "Value1",
    "OutputName2" : "Value2",
  }
}
```

3. Setelah mendapat SUCCESS respons, CloudFormation lanjutkan dengan operasi tumpukan. Jika respons FAILED atau tidak ada respons dikembalikan, operasi gagal. Data output apa pun dari sumber daya kustom disimpan di lokasi URL pre-signed. Pengembang template dapat mengambil data tersebut dengan menggunakan fungsi [Fn:: GetAtt](#).

Note

Jika Anda menggunakan AWS PrivateLink, sumber daya khusus di VPC harus memiliki akses ke bucket S3 CloudFormation -spesifik. Sumber daya kustom harus mengirim respons ke URL pre-signed Amazon S3. Jika mereka tidak dapat mengirim tanggapan ke Amazon S3, tidak CloudFormation akan menerima respons dan operasi tumpukan gagal. Untuk informasi selengkapnya, lihat [Akses CloudFormation menggunakan endpoint antarmuka \(\)AWS PrivateLink](#).

Batas waktu respons

Batas waktu default untuk sumber daya kustom Anda adalah 3600 detik (1 jam). Jika tidak ada respons yang diterima selama waktu ini, operasi tumpukan gagal.

Anda dapat menyesuaikan nilai batas waktu berdasarkan berapa lama Anda mengharapkan respons dari sumber daya kustom. Misalnya, saat menyediakan sumber daya khusus yang memanggil fungsi Lambda yang diharapkan merespons dalam waktu lima menit, Anda dapat mengatur batas waktu lima menit dalam template tumpukan dengan menentukan properti `ServiceTimeout`. Untuk informasi selengkapnya, lihat [Objek permintaan sumber daya kustom](#). Dengan cara ini, jika ada kesalahan dalam fungsi Lambda yang menyebabkannya macet, CloudFormation akan gagal operasi tumpukan setelah lima menit alih-alih menunggu satu jam penuh.

Namun, berhati-hatilah untuk tidak menetapkan nilai batas waktu terlalu rendah. Untuk menghindari batas waktu yang tidak terduga, pastikan sumber daya kustom Anda memiliki cukup waktu untuk melakukan tindakan yang diperlukan dan mengembalikan respons.

Sumber daya khusus yang didukung Amazon SNS

Topik berikut menunjukkan cara mengonfigurasi sumber daya kustom dengan token layanan yang menentukan topik Amazon SNS CloudFormation yang mengirimkan permintaan. Anda juga mempelajari urutan peristiwa dan pesan yang dikirim dan diterima sebagai hasil dari pembuatan, pembaruan, dan penghapusan tumpukan sumber daya kustom.

Dengan sumber daya khusus dan Amazon SNS, Anda dapat mengaktifkan skenario seperti menambahkan sumber daya baru ke tumpukan dan menyuntikkan data dinamis ke dalam tumpukan. Misalnya, saat Anda membuat tumpukan, CloudFormation dapat mengirim `Create` permintaan ke topik yang dipantau oleh aplikasi yang berjalan di EC2 instance Amazon. Notifikasi Amazon SNS memicu aplikasi untuk melakukan tugas penyediaan tambahan, seperti mengambil kumpulan alamat IP Elastis yang terdaftar yang diizinkan. Setelah selesai, aplikasi mengirimkan respons (dan data keluaran apa pun) yang memberi tahu CloudFormation untuk melanjutkan operasi tumpukan.

Saat Anda menentukan topik Amazon SNS sebagai target sumber daya kustom, CloudFormation mengirim pesan ke topik SNS yang ditentukan selama operasi tumpukan yang melibatkan sumber daya kustom. Untuk memproses pesan ini dan melakukan tindakan yang diperlukan, Anda harus memiliki titik akhir yang didukung berlangganan topik SNS.

Untuk pengenalan sumber daya khusus dan cara kerjanya, lihat [Buat logika penyediaan khusus dengan sumber daya khusus](#). Untuk informasi tentang Amazon SNS dan cara kerjanya, lihat Panduan [Pengembang Layanan Pemberitahuan Sederhana Amazon](#).

Menggunakan Amazon SNS untuk membuat sumber daya khusus

Topik

- [Langkah 1: Pembuatan tumpukan](#)
- [Langkah 2: Pembaruan tumpukan](#)
- [Langkah 3: Penghapusan tumpukan](#)

Langkah 1: Pembuatan tumpukan

1. Pengembang template membuat CloudFormation tumpukan yang berisi sumber daya khusus.

Dalam contoh template di bawah ini, kita menggunakan nama jenis sumber daya kustom `Custom::SeleniumTester` untuk sumber daya kustom dengan ID logis `MySeleniumTest`. Nama jenis sumber daya kustom harus alfanumerik dan dapat memiliki panjang maksimum 60 karakter.

Jenis sumber daya kustom dideklarasikan dengan token layanan, properti khusus penyedia opsional, dan `Fn::` atribut `GetAtt` opsional yang ditentukan oleh penyedia sumber daya kustom. Properti dan atribut ini dapat digunakan untuk meneruskan informasi dari pengembang template ke penyedia sumber daya khusus dan sebaliknya. Token layanan menentukan topik Amazon SNS yang telah dikonfigurasi oleh penyedia sumber daya.

```
{
  "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

Nama dan nilai data yang diakses `Fn::GetAtt` dikembalikan oleh penyedia sumber daya kustom selama respons penyedia CloudFormation. Jika penyedia sumber daya khusus adalah pihak ketiga, maka pengembang templat harus mendapatkan nama nilai pengembalian ini dari penyedia sumber daya khusus.

2. CloudFormation mengirimkan notifikasi Amazon SNS ke penyedia sumber daya dengan `"RequestType" : "Create"` yang berisi informasi tentang tumpukan, properti sumber daya khusus dari templat tumpukan, dan URL S3 untuk respons.

Topik SNS yang digunakan untuk mengirim notifikasi disematkan dalam template di `ServiceToken` properti. Untuk menghindari penggunaan nilai hardcode, pengembang template dapat menggunakan parameter template sehingga nilai dimasukkan pada saat tumpukan diluncurkan.

Contoh berikut menunjukkan permintaan `Create` sumber daya kustom yang mencakup nama jenis sumber daya kustom, `Custom::SeleniumTester`, yang dibuat dengan `LogicalResourceId` dari `MySeleniumTester`:

```

{
  "RequestType" : "Create",
  "ResponseURL" : "http://pre-signed-S3-url-for-response",
  "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-cd98-11ea-90d5-0a9cd3354c10",
  "RequestId" : "unique id for this create request",
  "ResourceType" : "Custom::SeleniumTester",
  "LogicalResourceId" : "MySeleniumTester",
  "ResourceProperties" : {
    "seleniumTester" : "SeleniumTest()",
    "endpoints" : [ "http://mysite.com", "http://myecommercesite.com/", "http://search.mysite.com" ],
    "frequencyOfTestsPerHour" : [ "3", "2", "4" ]
  }
}

```

```
}  
}
```

Untuk informasi rinci tentang objek permintaan untuk Create permintaan, lihat [Buat permintaan untuk sumber daya CloudFormation khusus](#) topik.

3. Penyedia sumber daya kustom memproses data yang dikirim oleh pengembang template dan menentukan apakah Create permintaan berhasil. Penyedia sumber daya kemudian menggunakan URL S3 yang dikirim oleh CloudFormation untuk mengirim respons salah satu SUCCESS atau FAILED.

Tergantung pada jenis respons, bidang respon yang berbeda akan diperkirakan oleh CloudFormation. Untuk informasi tentang bidang respons untuk jenis permintaan tertentu, lihat dokumentasi untuk jenis permintaan tersebut di [Tipe permintaan sumber daya khusus](#) bagian.

Menanggapi permintaan buat atau perbarui, penyedia sumber daya kustom dapat mengembalikan elemen data di [Data](#) bidang respons. Ini adalah pasangan nilai nama, dan nama sesuai dengan atribut Fn::GetAtt yang digunakan dengan sumber daya kustom dalam templat tumpukan. Nilai adalah data yang dikembalikan ketika pengembang template memanggil Fn::GetAtt sumber daya dengan nama atribut.

Berikut ini adalah contoh respons sumber daya kustom:

```
{  
  "Status" : "SUCCESS",  
  "PhysicalResourceId" : "Tester1",  
  "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-  
cd98-11ea-90d5-0a9cd3354c10",  
  "RequestId" : "unique id for this create request",  
  "LogicalResourceId" : "MySeleniumTester",  
  "Data" : {  
    "resultsPage" : "http://www.myexampledomain/test-results/guid",  
    "lastUpdate" : "2012-11-14T03:30Z"  
  }  
}
```

Untuk informasi rinci tentang objek respons untuk Create permintaan, lihat [Buat permintaan untuk sumber daya CloudFormation khusus](#) topiknya.

LogicalResourceIdBidang StackIdRequestId,, dan harus disalin kata demi kata dari permintaan.

4. CloudFormation mendeklarasikan status tumpukan sebagai `CREATE_COMPLETE` atau `CREATE_FAILED`. Jika tumpukan berhasil dibuat, pengembang template dapat menggunakan nilai output dari sumber daya kustom yang dibuat dengan mengaksesnya dengan `Fn::GetAtt`.

Sebagai contoh, templat sumber daya kustom yang digunakan untuk ilustrasi menggunakan `Fn::GetAtt` untuk menyalin output sumber daya ke dalam output tumpukan:

```
"Outputs" : {
  "topItem" : {
    "Value" : { "Fn::GetAtt" : ["MySeleniumTest", "resultsPage"] }
  },
  "numRespondents" : {
    "Value" : { "Fn::GetAtt" : ["MySeleniumTest", "lastUpdate"] }
  }
}
```

Langkah 2: Pembaruan tumpukan

Untuk memperbarui tumpukan yang ada, Anda harus mengirimkan templat yang menentukan pembaruan untuk properti sumber daya dalam tumpukan, seperti yang ditunjukkan pada contoh di bawah ini. CloudFormation hanya memperbarui sumber daya yang memiliki perubahan yang ditentukan dalam templat. Untuk informasi selengkapnya, lihat [Memahami perilaku pembaruan sumber daya tumpukan](#).

Anda dapat memperbarui sumber daya kustom yang memerlukan penggantian sumber daya fisik pokok. Saat Anda memperbarui sumber daya kustom dalam CloudFormation templat, CloudFormation kirimkan permintaan pembaruan ke sumber daya kustom tersebut. Jika sumber daya kustom memerlukan pengganti, sumber daya kustom baru harus mengirim respons dengan ID fisik baru. Saat CloudFormation menerima respons, itu membandingkan `PhysicalResourceId` antara sumber daya kustom lama dan baru. Jika berbeda, CloudFormation mengenali pembaruan sebagai pengganti dan mengirimkan permintaan hapus ke sumber daya lama, seperti yang ditunjukkan pada [Langkah 3: Penghapusan tumpukan](#).

Note

Jika Anda tidak membuat perubahan pada sumber daya kustom, CloudFormation tidak akan mengirim permintaan ke sana selama pembaruan tumpukan.

1. Pengembang template memulai pembaruan ke tumpukan yang berisi sumber daya khusus. Selama pembaruan, pengembang template dapat menentukan Properti baru di template tumpukan.

Berikut ini adalah contoh Update ke templat tumpukan menggunakan jenis sumber daya kustom:

```
{
  "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 mengirimkan pemberitahuan Amazon SNS ke penyedia sumber daya dengan "RequestType" : "Update" yang berisi informasi serupa dengan Create panggilan, kecuali bahwa OldResourceProperties bidang berisi properti sumber daya lama, dan ResourceProperties berisi properti sumber daya yang diperbarui (jika ada).

Berikut ini adalah contoh permintaan Update:

```
{
  "RequestType" : "Update",
  "ResponseURL" : "http://pre-signed-S3-url-for-response",
```

```

"StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-
cd98-11ea-90d5-0a9cd3354c10",
"RequestId" : "uniqueid for this update request",
"LogicalResourceId" : "MySeleniumTester",
"ResourceType" : "Custom::SeleniumTester",
"PhysicalResourceId" : "Tester1",
"ResourceProperties" : {
  "seleniumTester" : "SeleniumTest()",
  "endpoints" : [ "http://mysite.com", "http://myecommercesite.com/", "http://
search.mysite.com",
    "http://mynewsite.com" ],
  "frequencyOfTestsPerHour" : [ "3", "2", "4", "3" ]
},
"OldResourceProperties" : {
  "seleniumTester" : "SeleniumTest()",
  "endpoints" : [ "http://mysite.com", "http://myecommercesite.com/", "http://
search.mysite.com" ],
  "frequencyOfTestsPerHour" : [ "3", "2", "4" ]
}
}

```

Untuk informasi rinci tentang objek permintaan untuk Update permintaan, lihat [Permintaan pembaruan untuk sumber daya CloudFormation khusus](#) topik.

3. Penyedia sumber daya khusus memproses data yang dikirim oleh CloudFormation. Sumber daya kustom melakukan pembaruan dan mengirimkan respons dari salah satu SUCCESS atau FAILED ke URL S3. CloudFormation kemudian membandingkan sumber daya kustom lama dan baru. PhysicalResourceIDs Jika berbeda, CloudFormation mengakui bahwa pembaruan memerlukan penggantian dan mengirimkan permintaan hapus ke sumber daya lama. Contoh berikut menunjukkan respon penyedia sumber daya kustom untuk Update permintaan.

```

{
  "Status" : "SUCCESS",
  "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-
cd98-11ea-90d5-0a9cd3354c10",
  "RequestId" : "uniqueid for this update request",
  "LogicalResourceId" : "MySeleniumTester",
  "PhysicalResourceId" : "Tester2"
}

```

Untuk informasi rinci tentang objek respons untuk Update permintaan, lihat [Permintaan pembaruan untuk sumber daya CloudFormation khusus](#) topiknya.

`LogicalResourceId` `Bidang` `StackId` `RequestId`, dan harus disalin kata demi kata dari permintaan.

4. CloudFormation mendeklarasikan status tumpukan sebagai `UPDATE_COMPLETE` atau `UPDATE_FAILED`. Jika pembaruan gagal, tumpukan akan kembali. Jika tumpukan berhasil diperbarui, pengembang template dapat mengakses nilai keluaran baru dari sumber daya kustom yang dibuat dengan `Fn::GetAtt`.

Langkah 3: Penghapusan tumpukan

1. Pengembang template menghapus tumpukan yang berisi sumber daya khusus. CloudFormation mendapatkan properti saat ini yang ditentukan dalam template tumpukan bersama dengan topik SNS, dan bersiap untuk membuat permintaan ke penyedia sumber daya khusus.
2. CloudFormation mengirimkan notifikasi Amazon SNS ke penyedia sumber daya dengan `"RequestType" : "Delete"` yang berisi informasi terkini tentang tumpukan, properti sumber daya khusus dari templat tumpukan, dan URL S3 untuk respons.

Setiap kali Anda menghapus tumpukan atau membuat pembaruan yang menghapus atau menggantikan sumber daya kustom, CloudFormation bandingkan `PhysicalResourceId` antara sumber daya kustom lama dan baru. Jika berbeda, CloudFormation mengenali pembaruan sebagai pengganti dan mengirimkan permintaan hapus untuk sumber daya lama (`OldPhysicalResource`), seperti yang ditunjukkan pada contoh `Delete` permintaan berikut.

```
{
  "RequestType" : "Delete",
  "ResponseURL" : "http://pre-signed-S3-url-for-response",
  "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-
cd98-11ea-90d5-0a9cd3354c10",
  "RequestId" : "unique id for this delete request",
  "ResourceType" : "Custom::SeleniumTester",
  "LogicalResourceId" : "MySeleniumTester",
  "PhysicalResourceId" : "Tester1",
  "ResourceProperties" : {
    "seleniumTester" : "SeleniumTest()",
    "endpoints" : [ "http://mysite.com", "http://myecommercesite.com/", "http://
search.mysite.com",
    "http://mynewsite.com" ],
    "frequencyOfTestsPerHour" : [ "3", "2", "4", "3" ]
  }
}
```

```
}
```

Untuk informasi rinci tentang objek permintaan untuk Delete permintaan, lihat [Hapus permintaan untuk sumber daya CloudFormation khusus](#) topik.

DescribeStackResource, DescribeStackResources, dan ListStackResources menampilkan nama yang ditetapkan pengguna jika telah ditentukan.

3. Penyedia sumber daya kustom memproses data yang dikirim oleh CloudFormation dan menentukan apakah Delete permintaan berhasil. Penyedia sumber daya kemudian menggunakan URL S3 yang dikirim oleh CloudFormation untuk mengirim respons salah satu SUCCESS atau FAILED. Agar berhasil menghapus tumpukan dengan sumber daya kustom, penyedia sumber daya kustom harus berhasil merespons permintaan penghapusan.

Berikut ini adalah contoh respons penyedia sumber daya kustom terhadap Delete permintaan:

```
{
  "Status" : "SUCCESS",
  "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-
cd98-11ea-90d5-0a9cd3354c10",
  "RequestId" : "unique id for this delete request",
  "LogicalResourceId" : "MySeleniumTester",
  "PhysicalResourceId" : "Tester1"
}
```

Untuk informasi rinci tentang objek respons untuk Delete permintaan, lihat [Hapus permintaan untuk sumber daya CloudFormation khusus](#) topiknya.

LogicalResourceId Bidang StackIdRequestId,, dan harus disalin kata demi kata dari permintaan.

4. CloudFormation mendeklarasikan status tumpukan sebagai DELETE_COMPLETE atau DELETE_FAILED.

Sumber daya khusus yang didukung Lambda

Saat Anda mengaitkan fungsi Lambda dengan sumber daya khusus, fungsi tersebut akan dipanggil setiap kali sumber daya kustom dibuat, diperbarui, atau dihapus. CloudFormation memanggil API Lambda untuk memanggil fungsi dan meneruskan semua data permintaan (seperti jenis permintaan dan properti sumber daya) ke fungsi. Kekuatan dan kemampuan penyesuaian fungsi Lambda dalam

kombinasi dengan CloudFormation memungkinkan berbagai skenario, seperti mencari IDs AMI secara dinamis selama pembuatan tumpukan, atau mengimplementasikan dan menggunakan fungsi utilitas, seperti fungsi pembalikan string.

Untuk pengenalan sumber daya khusus dan cara kerjanya, lihat [Buat logika penyediaan khusus dengan sumber daya khusus](#).

Topik

- [Walkthrough: Buat mekanisme penundaan dengan sumber daya khusus yang didukung Lambda](#)
- [Modul cfn-response](#)

Walkthrough: Buat mekanisme penundaan dengan sumber daya khusus yang didukung Lambda

Panduan ini menunjukkan kepada Anda cara mengonfigurasi dan meluncurkan sumber daya kustom yang didukung Lambda menggunakan templat sampel. CloudFormation Template ini menciptakan mekanisme penundaan yang menjeda penerapan tumpukan untuk waktu yang ditentukan. Ini dapat berguna ketika Anda perlu memperkenalkan penundaan yang disengaja selama penyediaan sumber daya, seperti saat menunggu sumber daya stabil sebelum sumber daya dependen dibuat.

Note

Meskipun sumber daya kustom yang didukung Lambda sebelumnya direkomendasikan untuk mengambil IDs AMI, kami sekarang merekomendasikan penggunaan AWS Systems Manager parameter. Pendekatan ini membuat template Anda lebih dapat digunakan kembali dan lebih mudah dirawat. Untuk informasi selengkapnya, lihat [Dapatkan nilai plaintext dari Systems Manager Parameter Store](#).

Topik

- [Gambaran Umum](#)
- [Templat sampel](#)
- [Contoh panduan templat](#)
- [Prasyarat](#)
- [Meluncurkan tumpukan](#)
- [Membersihkan sumber daya](#)
- [Informasi terkait](#)

Gambaran Umum

Templat tumpukan sampel yang digunakan dalam panduan ini membuat sumber daya kustom yang didukung Lambda. Sumber daya khusus ini memperkenalkan penundaan yang dapat dikonfigurasi (60 detik secara default) selama pembuatan tumpukan. Penundaan terjadi selama pembaruan tumpukan hanya ketika properti sumber daya kustom diubah.

Template menyediakan sumber daya berikut:

- sumber daya khusus,
- fungsi Lambda, dan
- peran IAM yang memungkinkan Lambda untuk menulis log ke. CloudWatch

Ini juga mendefinisikan dua output:

- Waktu sebenarnya fungsi menunggu.
- Pengidentifikasi unik yang dihasilkan selama setiap eksekusi fungsi Lambda.

Note

CloudFormation adalah layanan gratis tetapi Lambda mengenakan biaya berdasarkan jumlah permintaan untuk fungsi Anda dan waktu kode Anda dijalankan. [Untuk informasi selengkapnya tentang harga Lambda, lihat AWS Lambda harga.](#)

Templat sampel

Anda dapat melihat template sampel sumber daya kustom yang didukung Lambda dengan mekanisme penundaan di bawah ini:

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Resources": {
    "LambdaExecutionRole": {
      "Type": "AWS::IAM::Role",
      "Properties": {
        "AssumeRolePolicyDocument": {
```

```

        "Statement": [{
            "Effect": "Allow",
            "Principal": { "Service": ["lambda.amazonaws.com"] },
            "Action": ["sts:AssumeRole"]
        }]
    },
    "Path": "/",
    "Policies": [{
        "PolicyName": "AllowLogs",
        "PolicyDocument": {
            "Statement": [{
                "Effect": "Allow",
                "Action": ["logs:*"],
                "Resource": "*"
            }]
        }
    }]
}
},
"CFNWaiter": {
    "Type": "AWS::Lambda::Function",
    "Properties": {
        "Handler": "index.handler",
        "Runtime": "python3.9",
        "Timeout": 900,
        "Role": { "Fn::GetAtt": ["LambdaExecutionRole", "Arn"] },
        "Code": {
            "ZipFile": { "Fn::Join": ["\n", [
                "from time import sleep",
                "import json",
                "import cfnresponse",
                "import uuid",
                "",
                "def handler(event, context):",
                "    wait_seconds = 0",
                "    id = str(uuid.uuid1())",
                "    if event[\"RequestType\"] in [\"Create\", \"Update\"]:",
                "        wait_seconds = int(event[\"ResourceProperties\"].get(\"WaitSeconds\",
0))),
                "        sleep(wait_seconds)",
                "    response = {",
                "        \"TimeWaited\": wait_seconds,",
                "        \"Id\": id ",
                "    }",
            ]}
        }
    }
}

```

```

        " cfnresponse.send(event, context, cfnresponse.SUCCESS, response,
        \"Waiter-\"+id)\"
        ]}]
    }
}
},
\"CFNWaiterCustomResource\": {
    \"Type\": \"AWS::CloudFormation::CustomResource\",
    \"Properties\": {
        \"ServiceToken\": { \"Fn::GetAtt\": [\"CFNWaiter\", \"Arn\"] },
        \"WaitSeconds\": 60
    }
}
},
\"Outputs\": {
    \"TimeWaited\": {
        \"Value\": { \"Fn::GetAtt\": [\"CFNWaiterCustomResource\", \"TimeWaited\"] },
        \"Export\": { \"Name\": \"TimeWaited\" }
    },
    \"WaiterId\": {
        \"Value\": { \"Fn::GetAtt\": [\"CFNWaiterCustomResource\", \"Id\"] },
        \"Export\": { \"Name\": \"WaiterId\" }
    }
}
}
}

```

YAML

```

AWSTemplateFormatVersion: \"2010-09-09\"
Resources:
  LambdaExecutionRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Statement:
          - Effect: \"Allow\"
            Principal:
              Service:
                - \"lambda.amazonaws.com\"
            Action:
              - \"sts:AssumeRole\"
      Path: \"/\"
    Policies:

```

```

    - PolicyName: "AllowLogs"
      PolicyDocument:
        Statement:
          - Effect: "Allow"
            Action:
              - "logs:*"
            Resource: "*"
  CFNWaiter:
    Type: AWS::Lambda::Function
    Properties:
      Handler: index.handler
      Runtime: python3.9
      Timeout: 900
      Role: !GetAtt LambdaExecutionRole.Arn
      Code:
        ZipFile:
          !Sub |
            from time import sleep
            import json
            import cfnresponse
            import uuid

            def handler(event, context):
                wait_seconds = 0
                id = str(uuid.uuid1())
                if event["RequestType"] in ["Create", "Update"]:
                    wait_seconds = int(event["ResourceProperties"].get("WaitSeconds", 0))
                    sleep(wait_seconds)
                response = {
                    "TimeWaited": wait_seconds,
                    "Id": id
                }
                cfnresponse.send(event, context, cfnresponse.SUCCESS, response,
"Waiter-"+id)
  CFNWaiterCustomResource:
    Type: "AWS::CloudFormation::CustomResource"
    Properties:
      ServiceToken: !GetAtt CFNWaiter.Arn
      WaitSeconds: 60
Outputs:
  TimeWaited:
    Value: !GetAtt CFNWaiterCustomResource.TimeWaited
  Export:
    Name: TimeWaited

```

```
WaiterId:
  Value: !GetAtt CFNWaiterCustomResource.Id
  Export:
    Name: WaiterId
```

Contoh panduan templat

Cuplikan berikut menjelaskan bagian yang relevan dari template sampel untuk membantu Anda memahami bagaimana fungsi Lambda dikaitkan dengan sumber daya khusus dan memahami hasilnya.

[AWS::Lambda::Function](#) sumber daya CFNWaiter

`AWS::Lambda::FunctionResource` menentukan kode sumber fungsi, nama handler, lingkungan runtime, dan peran eksekusi Amazon Resource Name (ARN).

`HandlerProperti` diatur ke `index.handler` karena menggunakan kode sumber Python. [Untuk informasi selengkapnya tentang pengidentifikasi handler yang diterima saat menggunakan kode sumber fungsi inline, lihat Kode. `AWS::Lambda::Function`](#)

`Runtime` ini ditentukan sebagai `python3.9` karena file sumber adalah kode Python.

`Timeout` ini diatur ke 900 detik.

`RoleProperti` menggunakan `Fn::GetAtt` fungsi untuk mendapatkan ARN dari peran `LambdaExecutionRole` eksekusi yang dideklarasikan dalam `AWS::IAM::Role` sumber daya dalam template.

`CodeProperti` mendefinisikan kode fungsi inline menggunakan fungsi Python. Fungsi Python dalam template sampel melakukan hal berikut:

- Buat ID unik menggunakan UUID
- Periksa apakah permintaan tersebut adalah permintaan buat atau perbarui
- Tidur selama durasi yang ditentukan `WaitSeconds` selama `Create` atau `Update` permintaan
- Kembalikan waktu tunggu dan ID unik

JSON

```
...
  "CFNWaiter": {
```

```

    "Type": "AWS::Lambda::Function",
    "Properties": {
        "Handler": "index.handler",
        "Runtime": "python3.9",
        "Timeout": 900,
        "Role": { "Fn::GetAtt": ["LambdaExecutionRole", "Arn"] },
        "Code": {
            "ZipFile": { "Fn::Join": ["\n", [
                "from time import sleep",
                "import json",
                "import cfnresponse",
                "import uuid",
                "",
                "def handler(event, context):",
                "    wait_seconds = 0",
                "    id = str(uuid.uuid1())",
                "    if event[\"RequestType\"] in [\"Create\", \"Update\"]:",
                "        wait_seconds = int(event[\"ResourceProperties\"].get(\"WaitSeconds\",
0))),
                "        sleep(wait_seconds)",
                "        response = {",
                "            \"TimeWaited\": wait_seconds,",
                "            \"Id\": id ",
                "        }",
                "        cfnresponse.send(event, context, cfnresponse.SUCCESS, response,
\"Waiter-\"+id)
            ]]}
        ]
    },
    ...

```

YAML

```

...
CFNWaiter:
  Type: AWS::Lambda::Function
  Properties:
    Handler: index.handler
    Runtime: python3.9
    Timeout: 900
    Role: !GetAtt LambdaExecutionRole.Arn
    Code:

```

```

ZipFile:
    !Sub |
    from time import sleep
    import json
    import cfnresponse
    import uuid

    def handler(event, context):
        wait_seconds = 0
        id = str(uuid.uuid1())
        if event["RequestType"] in ["Create", "Update"]:
            wait_seconds = int(event["ResourceProperties"].get("WaitSeconds", 0))
            sleep(wait_seconds)
        response = {
            "TimeWaited": wait_seconds,
            "Id": id
        }
        cfnresponse.send(event, context, cfnresponse.SUCCESS, response,
            "Waiter-"+id)
    ...

```

[AWS::IAM::Role](#) sumber daya LambdaExecutionRole

AWS::IAM:Role Sumber daya menciptakan peran eksekusi untuk fungsi Lambda, yang mencakup kebijakan peran asumsi yang memungkinkan Lambda menggunakannya. Ini juga berisi kebijakan yang memungkinkan akses CloudWatch Log.

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](#) sumber daya CFNWaiterCustomResource

Sumber daya khusus menautkan ke fungsi Lambda dengan penggunaan ARN-nya. !GetAtt CFNWaiter.Arn Ini akan menerapkan waktu tunggu 60 detik untuk membuat dan memperbarui

operasi, seperti yang diatur `WaitSeconds`. Sumber daya hanya akan dipanggil untuk operasi pembaruan jika properti diubah.

JSON

```
...
  "CFNWaiterCustomResource": {
    "Type": "AWS::CloudFormation::CustomResource",
    "Properties": {
      "ServiceToken": { "Fn::GetAtt": ["CFNWaiter", "Arn"] },
      "WaitSeconds": 60
    }
  },
  ...
```

YAML

```
...
CFNWaiterCustomResource:
  Type: "AWS::CloudFormation::CustomResource"
  Properties:
    ServiceToken: !GetAtt CFNWaiter.Arn
    WaitSeconds: 60
...
```

Outputs

Template ini adalah `TimeWaited` dan `WaiterId`. Outputs `TimeWaited` Nilai menggunakan `Fn::GetAtt` fungsi untuk memberikan jumlah waktu sumber daya pelayan benar-benar menunggu. `WaiterId` Menggunakan `Fn::GetAtt` fungsi untuk memberikan ID unik yang dihasilkan dan terkait dengan eksekusi.

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  
...  
}
```

Prasyarat

Anda harus memiliki izin IAM untuk menggunakan semua layanan terkait, seperti Lambda dan CloudFormation

Meluncurkan tumpukan

Untuk membuat tumpukan

1. Temukan template preferensi Anda (YAMAL atau JSON) dari [Templat sampel](#) bagian dan simpan ke mesin Anda dengan nama. `samplelambdabackedcustomresource.template`
2. Buka CloudFormation konsol di <https://console.aws.amazon.com/cloudformation/>.
3. Dari halaman Stacks, pilih Buat tumpukan di kanan atas, lalu pilih Dengan sumber daya baru (standar).
4. Untuk Prasyarat - Siapkan template, pilih Pilih template yang ada.
5. Untuk Menentukan templat, pilih Unggah file templat, lalu pilih Pilih file.

6. Pilih file `samplelambdabackedcustomresource.template` template yang Anda simpan sebelumnya.
7. Pilih Berikutnya.
8. Untuk nama Stack, ketik **SampleCustomResourceStack** dan pilih Berikutnya.
9. Untuk panduan ini, Anda tidak perlu menambahkan tanda atau menentukan pengaturan lanjutan, jadi pilih Selanjutnya.
10. Pastikan nama tumpukan terlihat benar, lalu pilih Buat.

Mungkin perlu beberapa menit CloudFormation untuk membuat tumpukan Anda. Untuk memantau kemajuan, tampilkan peristiwa tumpukan. Untuk informasi selengkapnya, lihat [Lihat informasi tumpukan dari CloudFormation konsol](#).

Jika pembuatan tumpukan berhasil, semua sumber daya dalam tumpukan, seperti fungsi Lambda dan sumber daya khusus, dibuat. Anda telah berhasil menggunakan fungsi Lambda dan sumber daya khusus.

[Jika fungsi Lambda mengembalikan kesalahan, lihat log fungsi di konsol CloudWatch Log.](#)

Nama pengaliran log adalah ID fisik sumber daya kustom, yang dapat Anda temukan dengan melihat sumber daya tumpukan. Untuk informasi selengkapnya, [lihat Melihat data log](#) di Panduan CloudWatch Pengguna Amazon.

Membersihkan sumber daya

Hapus tumpukan untuk membersihkan semua sumber daya tumpukan yang Anda buat sehingga Anda tidak dikenakan biaya untuk sumber daya yang tidak diperlukan.

Untuk menghapus tumpukan

1. Dari CloudFormation konsol, pilih SampleCustomResourceStacktumpukan.
2. Pilih Tindakan, dan kemudian Hapus Tumpukan.
3. Pada pesan konfirmasi, pilih Ya, Hapus.

Semua sumber daya yang telah Anda buat sebelumnya akan dihapus.

Sekarang setelah Anda memahami cara membuat dan menggunakan sumber daya kustom yang didukung Lambda, Anda dapat menggunakan contoh template dan kode dari panduan ini untuk membangun dan bereksperimen dengan tumpukan dan fungsi lain.

Informasi terkait

- [CloudFormation Referensi Sumber Daya Kustom](#)
- [AWS::CloudFormation::CustomResource](#)

Modul **cfn-response**

Dalam CloudFormation template Anda, Anda dapat menentukan fungsi Lambda sebagai target sumber daya kustom. Saat Anda menggunakan `ZipFile` properti untuk menentukan kode sumber [fungsi](#) Anda, Anda dapat memuat `cfn-response` modul untuk mengirim respons dari fungsi Lambda Anda ke sumber daya khusus. `cfn-response` Modul ini adalah pustaka yang menyederhanakan pengiriman tanggapan ke sumber daya kustom yang memanggil fungsi Lambda Anda. Modul ini memiliki `send` metode yang mengirimkan [objek respons](#) ke sumber daya khusus melalui URL presigned Amazon S3 (the). `ResponseURL`

Modul `cfn-response` tersedia hanya ketika Anda menggunakan properti `ZipFile` untuk menulis kode sumber Anda. Ini tidak tersedia untuk kode sumber yang disimpan di bucket Amazon S3. Untuk kode dalam bucket, Anda harus menulis fungsi Anda sendiri untuk mengirim tanggapan.

Note

Setelah mengeksekusi `send` metode, fungsi Lambda berakhir, jadi apa pun yang Anda tulis setelah metode itu diabaikan.

Memuat modul **cfn-response**

Untuk fungsi Node.js, gunakan fungsi `require()` untuk memuat modul `cfn-response`. Sebagai contoh, contoh kode berikut membuat objek `cfn-response` dengan nama `response`:

```
var response = require('cfn-response');
```

Untuk Python, gunakan pernyataan `import` untuk memuat modul `cfnresponse`, seperti yang ditunjukkan dalam contoh berikut:

Note

Gunakan pernyataan impor persis seperti ini. Jika Anda menggunakan varian lain dari pernyataan impor, CloudFormation tidak mencakup modul respon.

```
import cfnresponse
```

Parameter metode `send`

Anda dapat menggunakan parameter berikut dengan metode `send`.

`event`

Bidang dalam [permintaan sumber daya kustom](#).

`context`

Objek, khusus untuk fungsi Lambda, yang dapat Anda gunakan untuk menentukan kapan fungsi dan callback telah menyelesaikan eksekusi, atau untuk mengakses informasi dari dalam lingkungan eksekusi Lambda. Untuk informasi selengkapnya, lihat [Membangun fungsi Lambda dengan Node.js](#) di Panduan AWS Lambda Pengembang.

`responseStatus`

Apakah fungsi berhasil diselesaikan. Gunakan konstanta modul `cfnresponse` untuk menentukan status: `SUCCESS` untuk eksekusi yang sukses dan `FAILED` untuk eksekusi yang gagal.

`responseData`

Bidang Data dari [objek respons](#) sumber daya kustom. Data tersebut merupakan daftar pasangan nama-nilai.

`physicalResourceId`

Opsional. Pengidentifikasi unik dari sumber daya kustom yang memanggil fungsi tersebut. Secara default, modul menggunakan nama aliran CloudWatch log Amazon Logs yang terkait dengan fungsi Lambda.

Nilai yang dikembalikan untuk `PhysicalResourceId` dapat mengubah operasi pembaruan sumber daya kustom. Jika nilai yang dikembalikan sama, itu dianggap sebagai pembaruan normal. Jika nilai yang dikembalikan berbeda, CloudFormation mengenali pembaruan sebagai

pengganti dan mengirimkan permintaan hapus ke sumber daya lama. Untuk informasi selengkapnya, lihat [AWS::CloudFormation::CustomResource](#).

noEcho

Opsional. Menunjukkan apakah menutupi output dari sumber daya kustom ketika ia diambil dengan menggunakan fungsi `Fn::GetAtt`. Jika diatur ke `true`, semua nilai yang dikembalikan ditutupi dengan tanda bintang (`*****`), kecuali untuk informasi yang tersimpan di lokasi yang ditentukan di bawah ini. Secara default, nilainya adalah `false`.

Important

Menggunakan NoEcho atribut tidak menutupi informasi apa pun yang disimpan sebagai berikut:

- Bagian Metadata template. CloudFormation tidak mengubah, memodifikasi, atau menyunting informasi apa pun yang Anda sertakan di Metadata bagian ini. Untuk informasi selengkapnya, lihat [Metadata](#).
- Bagian Outputs template. Untuk informasi selengkapnya, lihat [Output](#).
- MetadataAtribut definisi sumber daya. Untuk informasi lebih lanjut, lihat [Atribut Metadata](#).

Kami sangat menyarankan Anda untuk tidak menggunakan mekanisme ini untuk memasukkan informasi sensitif, seperti kata sandi atau rahasia.

Untuk informasi selengkapnya tentang penggunaan NoEcho untuk menutupi informasi sensitif, lihat praktik [Jangan menanamkan kredensial dalam templat Anda](#) terbaik.

Contoh

Node.js

Dalam contoh Node.js berikut, fungsi Lambda inline mengambil nilai input dan mengalikannya dengan 5. Fungsi inline sangat berguna untuk fungsi yang lebih kecil karena memungkinkan Anda menentukan kode sumber secara langsung di template, alih-alih membuat paket dan mengunggahnya ke bucket Amazon S3. Fungsi ini menggunakan metode `cfn-response send` untuk mengirim hasil kembali ke sumber daya kustom yang memanggilnya.

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

Dalam contoh Python berikut, fungsi Lambda inline mengambil nilai integer dan mengalikannya dengan 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")
```

Kode sumber modul

Topik

- [Kode sumber Node.js asinkron](#)
- [Kode sumber Node.js](#)
- [Kode sumber Python](#)

Kode sumber Node.js asinkron

Berikut ini adalah kode sumber modul respons untuk fungsi Node.js jika handler asinkron. Tinjau untuk memahami apa yang dilakukan modul dan untuk bantuan dengan menerapkan fungsi respon Anda sendiri.

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: MIT-0

exports.SUCCESS = "SUCCESS";
exports.FAILED = "FAILED";

exports.send = function(event, context, responseStatus, responseData,
    physicalResourceId, noEcho) {

    return new Promise((resolve, reject) => {
        var responseBody = JSON.stringify({
            Status: responseStatus,
            Reason: "See the details in CloudWatch Log Stream: " +
context.logStreamName,
            PhysicalResourceId: physicalResourceId || context.logStreamName,
            StackId: event.StackId,
            RequestId: event.RequestId,
            LogicalResourceId: event.LogicalResourceId,
            NoEcho: noEcho || false,
```



```
        Data: responseData
    });

    console.log("Response body:\n", responseBody);

    var https = require("https");
    var url = require("url");

    var parsedUrl = url.parse(event.ResponseURL);
    var options = {
        hostname: parsedUrl.hostname,
        port: 443,
        path: parsedUrl.path,
        method: "PUT",
        headers: {
            "content-type": "",
            "content-length": responseBody.length
        }
    };

    var request = https.request(options, function(response) {
        console.log("Status code: " + parseInt(response.statusCode));
        resolve(context.done());
    });

    request.on("error", function(error) {
        console.log("send(..) failed executing https.request(..): " +
maskCredentialsAndSignature(error));
        reject(context.done(error));
    });

    request.write(responseBody);
    request.end();
})
}

function maskCredentialsAndSignature(message) {
    return message.replace(/X-Amz-Credential=[^\s]+/i, 'X-Amz-Credential=*****')
        .replace(/X-Amz-Signature=[^\s]+/i, 'X-Amz-Signature=*****');
}
```

Kode sumber Node.js

Berikut ini adalah kode sumber modul respons untuk fungsi Node.js jika handler tidak asinkron. Tinjau untuk memahami apa yang dilakukan modul dan untuk bantuan dengan menerapkan fungsi respon Anda sendiri.

```
// 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();
}

```

Kode sumber Python

Berikut ini adalah kode sumber modul respon untuk fungsi Python:

```

# 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)

```

Referensi sumber daya kustom

Untuk pengenalan sumber daya khusus dan cara kerjanya, lihat [Buat logika penyediaan khusus dengan sumber daya khusus](#).

Bagian ini menyediakan detail tentang:

- Bidang permintaan dan respons JSON yang digunakan dalam pesan yang dikirim ke dan dari CloudFormation saat menyediakan sumber daya kustom.

- Bidang yang diharapkan untuk permintaan, dan tanggapan terhadap, penyedia sumber daya khusus sebagai respons terhadap pembuatan tumpukan, pembaruan tumpukan, dan penghapusan tumpukan.

Topik

- [Objek permintaan sumber daya kustom](#)
- [Objek tanggapan sumber daya khusus](#)
- [Tipe permintaan sumber daya khusus](#)

Objek permintaan sumber daya kustom

Topik ini menjelaskan properti objek permintaan untuk sumber daya CloudFormation kustom.

Untuk pengenalan sumber daya khusus dan cara kerjanya, lihat [Buat logika penyediaan khusus dengan sumber daya khusus](#).

Properti permintaan developer templat

Pengembang template menggunakan CloudFormation sumber daya [AWS::CloudFormation::CustomResource](#), untuk menentukan sumber daya kustom dalam template.

Di `AWS::CloudFormation::CustomResource`, semua properti ditentukan oleh penyedia sumber daya kustom. Hanya ada satu properti yang diperlukan: `ServiceToken`.

ServiceTimeout

Waktu maksimum, dalam hitungan detik, yang dapat berlalu sebelum waktu operasi sumber daya kustom habis.

Nilai harus berupa bilangan bulat dari 1 hingga 3600. Nilai default adalah 3600 detik (1 jam).

Wajib: Tidak

Tipe: String

ServiceToken

Token layanan, seperti topik Amazon SNS ARN atau fungsi Lambda ARN. Token layanan harus dari Wilayah yang sama dengan tumpukan.

Wajib: Ya

Tipe: String

Semua bidang lain di properti sumber daya bersifat opsional dan dikirim, kata demi kata, ke penyedia sumber daya kustom di bidang permintaan. `ResourceProperties` Penyedia menentukan nama maupun konten yang valid dari bidang ini.

Bidang permintaan penyedia sumber daya khusus

Bidang ini dikirim dalam permintaan JSON dari CloudFormation penyedia sumber daya khusus dalam topik SNS yang telah dikonfigurasi penyedia untuk tujuan ini.

`RequestType`

Jenis permintaan diatur oleh operasi CloudFormation tumpukan (create-stack, update-stack, atau delete-stack) yang diprakarsai oleh pengembang template untuk tumpukan yang berisi sumber daya kustom.

Harus salah satu dari: `Create`, `Update`, atau `Delete`. Untuk informasi selengkapnya, lihat [Tipe permintaan sumber daya khusus](#).

Wajib: Ya

Tipe: String

`ResponseURL`

URL respons mengidentifikasi bucket S3 yang telah ditetapkan sebelumnya yang menerima tanggapan dari penyedia sumber daya kustom ke. AWS CloudFormation

Wajib: Ya

Tipe: String

`StackId`

Nama Sumber Daya Amazon (ARN) yang mengidentifikasi tumpukan yang berisi sumber daya khusus.

Menggabungkan `StackId` dengan `RequestId` formulir nilai yang dapat Anda gunakan untuk mengidentifikasi permintaan secara unik pada sumber daya khusus tertentu.

Wajib: Ya

Tipe: String

`RequestId`

ID unik untuk permintaan tersebut.

Menggabungkan `StackId` dengan `RequestId` formulir nilai yang dapat Anda gunakan untuk mengidentifikasi permintaan secara unik pada sumber daya khusus tertentu.

Wajib: Ya

Tipe: String

`ResourceType`

Jenis sumber daya yang dipilih developer templat dari sumber daya kustom di templat CloudFormation . Nama jenis sumber daya kustom dapat memiliki panjang hingga 60 karakter dan dapat mencakup alfanumerik dan karakter berikut: `_@-`.

Wajib: Ya

Tipe: String

`LogicalResourceId`

Nama template yang dipilih pengembang (ID logis) dari sumber daya kustom dalam template. AWS CloudFormation ini disediakan untuk memfasilitasi komunikasi antara penyedia sumber daya khusus dan pengembang template.

Wajib: Ya

Tipe: String

`PhysicalResourceId`

ID fisik yang ditentukan penyedia sumber daya kustom wajib yang unik untuk penyedia tersebut.

Nilai yang dikembalikan untuk `PhysicalResourceId` dapat mengubah operasi pembaruan sumber daya kustom. Jika nilai yang dikembalikan sama, itu dianggap sebagai pembaruan normal. Jika nilai yang dikembalikan berbeda, AWS CloudFormation mengenali pembaruan sebagai pengganti dan mengirimkan permintaan hapus ke sumber daya lama. Untuk informasi selengkapnya, lihat [AWS::CloudFormation::CustomResource](#).

Wajib: Selalu dikirim dengan Update dan Delete permintaan; tidak pernah dikirim dengan Create.

Tipe: String

ResourceProperties

Bidang ini berisi isi Properties objek yang dikirim oleh pengembang template. Isinya ditentukan oleh penyedia sumber daya khusus.

Wajib: Tidak

Jenis: objek JSON

OldResourceProperties

Digunakan hanya untuk Update permintaan. Berisi properti sumber daya yang dideklarasikan sebelum permintaan pembaruan.

Wajib: Ya

Jenis: objek JSON

Objek tanggapan sumber daya khusus

Topik ini menjelaskan properti objek respons untuk sumber daya CloudFormation kustom.

Untuk pengenalan sumber daya khusus dan cara kerjanya, lihat [Buat logika penyediaan khusus dengan sumber daya khusus](#).

Bidang respons penyedia sumber daya khusus

Berikut ini adalah properti yang disertakan penyedia sumber daya khusus saat mengirim file JSON ke URL yang telah ditandatangani. Untuk informasi selengkapnya tentang mengunggah objek menggunakan presigned URLs, lihat [Mengunggah objek dengan presigned di Panduan Pengguna URLs](#) Layanan Penyimpanan Sederhana Amazon.

Note

Ukuran total badan respons tidak dapat melebihi 4096 byte.

Status

Nilai status yang dikirim oleh penyedia sumber daya kustom sebagai respons terhadap permintaan AWS CloudFormation yang dihasilkan.

Harus SUCCESS atau FAILED.

Wajib: Ya

Tipe: String

Reason

Menjelaskan alasan respons kegagalan.

Diperlukan: Diperlukan jika Status ada FAILED. Sebaliknya, ini bersifat opsional.

Tipe: String

PhysicalResourceId

Nilai ini harus menjadi pengenalan yang unik untuk vendor sumber daya khusus, dan dapat berukuran hingga 1 KB. Nilai harus berupa string yang tidak kosong dan harus identik untuk semua tanggapan untuk sumber daya yang sama.

Nilai yang dikembalikan untuk PhysicalResourceId dapat mengubah operasi pembaruan sumber daya kustom. Jika nilai yang dikembalikan sama, itu dianggap sebagai pembaruan normal. Jika nilai yang dikembalikan berbeda, AWS CloudFormation mengenali pembaruan sebagai pengganti dan mengirimkan permintaan hapus ke sumber daya lama. Untuk informasi selengkapnya, lihat [AWS::CloudFormation::CustomResource](#).

Wajib: Ya

Tipe: String

StackId

Nama Sumber Daya Amazon (ARN) yang mengidentifikasi tumpukan yang berisi sumber daya khusus. Nilai respons ini harus disalin kata demi kata dari permintaan.

Wajib: Ya

Tipe: String

RequestId

ID unik untuk permintaan tersebut. Nilai respons ini harus disalin kata demi kata dari permintaan.

Wajib: Ya

Tipe: String

LogicalResourceId

Nama template yang dipilih pengembang (ID logis) dari sumber daya kustom dalam template. AWS CloudFormation Nilai respons ini harus disalin kata demi kata dari permintaan.

Wajib: Ya

Tipe: String

NoEcho

Opsional. Menunjukkan apakah akan menutupi output dari sumber daya kustom saat diambil dengan menggunakan Fn::GetAtt fungsi. Jika disetel ke true, semua nilai yang dikembalikan ditutupi dengan tanda bintang (*****), kecuali yang disimpan di bagian *Metadata* template. AWS CloudFormation tidak mengubah, memodifikasi, atau menyunting informasi apa pun yang Anda sertakan di Metadata bagian ini. Nilai default-nya adalah false.

Untuk informasi lebih lanjut tentang menggunakan NoEcho untuk menutupi informasi sensitif, lihat praktik terbaik [Jangan sematkan kredensial dalam templat Anda](#).

Wajib: Tidak

Jenis: Boolean

Data

Opsional. Pasangan nama-nilai yang ditentukan penyedia sumber daya khusus untuk dikirim dengan respons. Anda dapat mengakses nilai yang disediakan di sini dengan nama di template dengan Fn::GetAtt.

Important

Jika pasangan nama-nilai berisi informasi sensitif, Anda harus menggunakan NoEcho bidang untuk menutupi output dari sumber daya kustom. Jika tidak, nilai terlihat melalui APIs nilai properti permukaan (seperti DescribeStackEvents).

Wajib: Tidak

Jenis: objek JSON

Tipe permintaan sumber daya khusus

Jenis permintaan dikirim di RequestType bidang dalam [objek permintaan](#) yang dikirim oleh AWS CloudFormation saat pengembang template membuat, memperbarui, atau menghapus tumpukan yang berisi sumber daya kustom.

Setiap jenis permintaan memiliki kumpulan bidang tertentu yang dikirim bersama permintaan, termasuk URL Amazon S3 untuk respons oleh penyedia sumber daya kustom. Penyedia harus menanggapi bucket S3 dengan FAILED hasil SUCCESS atau hasil sebelum periode batas waktu berakhir. Jika tidak ada respons yang diterima sebelum periode batas waktu berakhir, permintaan akan dianggap tidak berhasil dan operasi tumpukan gagal. Setiap hasil juga memiliki seperangkat bidang tertentu yang diharapkan oleh CloudFormation.

Bagian ini menyediakan informasi tentang bidang permintaan dan respons, dengan contoh, untuk setiap tipe permintaan.

Topik

- [Buat permintaan untuk sumber daya CloudFormation khusus](#)
- [Hapus permintaan untuk sumber daya CloudFormation khusus](#)
- [Permintaan pembaruan untuk sumber daya CloudFormation khusus](#)

Buat permintaan untuk sumber daya CloudFormation khusus

Saat pengembang template membuat tumpukan yang berisi sumber daya khusus, CloudFormation kirimkan permintaan ke penyedia sumber daya khusus dengan RequestType disetel keCreate. Permintaan ini terjadi secara khusus ketika sumber daya khusus sedang dibuat.

Untuk pengenalan sumber daya khusus dan cara kerjanya, lihat[Buat logika penyediaan khusus dengan sumber daya khusus](#).

Permintaan

Buat permintaan berisi bidang berikut:

RequestType

Create.

RequestId

ID unik untuk permintaan tersebut.

ResponseURL

URL respons mengidentifikasi bucket S3 yang telah ditetapkan sebelumnya yang menerima tanggapan dari penyedia sumber daya kustom ke. AWS CloudFormation

ResourceType

Jenis sumber daya yang dipilih developer templat dari sumber daya kustom di templat CloudFormation . Nama jenis sumber daya kustom dapat memiliki panjang hingga 60 karakter dan dapat mencakup alfanumerik dan karakter berikut: `_@-`.

LogicalResourceId

Nama template yang dipilih pengembang (ID logis) dari sumber daya kustom dalam template. AWS CloudFormation

StackId

Nama Sumber Daya Amazon (ARN) yang mengidentifikasi tumpukan yang berisi sumber daya khusus.

ResourceProperties

Bidang ini berisi isi Properties objek yang dikirim oleh pengembang template. Isinya ditentukan oleh penyedia sumber daya khusus.

Contoh

```
{
  "RequestType" : "Create",
  "RequestId" : "unique id for this create request",
  "ResponseURL" : "pre-signed-url-for-create-response",
  "ResourceType" : "Custom::MyCustomResourceType",
  "LogicalResourceId" : "name of resource in template",
  "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-cd98-11ea-90d5-0a9cd3354c10",
  "ResourceProperties" : {
    "key1" : "string",
    "key2" : [ "list" ],
    "key3" : { "key4" : "map" }
```

```
}  
}
```

Respons

Berhasil

Jika permintaan buat berhasil, respons harus dikirim ke bucket Amazon S3 dengan bidang berikut:

Status

Harus berupa SUCCESS.

RequestId

ID unik untuk permintaan tersebut. Nilai respons ini harus disalin kata demi kata dari permintaan.

LogicalResourceId

Nama template yang dipilih pengembang (ID logis) dari sumber daya kustom dalam template. AWS CloudFormation Nilai respons ini harus disalin kata demi kata dari permintaan.

StackId

Nama Sumber Daya Amazon (ARN) yang mengidentifikasi tumpukan yang berisi sumber daya khusus. Nilai respons ini harus disalin kata demi kata dari permintaan.

PhysicalResourceId

Nilai ini harus menjadi pengenalan yang unik untuk vendor sumber daya khusus, dan dapat berukuran hingga 1 KB. Nilai harus berupa string yang tidak kosong dan harus identik untuk semua tanggapan untuk sumber daya yang sama.

Nilai yang dikembalikan untuk `PhysicalResourceId` dapat mengubah operasi pembaruan sumber daya kustom. Jika nilai yang dikembalikan sama, itu dianggap sebagai pembaruan normal. Jika nilai yang dikembalikan berbeda, AWS CloudFormation mengenali pembaruan sebagai pengganti dan mengirimkan permintaan hapus ke sumber daya lama. Untuk informasi selengkapnya, lihat [AWS::CloudFormation::CustomResource](#).

NoEcho

Opsional. Menunjukkan apakah akan menutupi output dari sumber daya kustom saat diambil dengan menggunakan `Fn::GetAtt` fungsi. Jika disetel ke `true`, semua nilai yang dikembalikan ditutupi dengan tanda bintang (`*****`), kecuali yang disimpan di bagian *Metadata* template. AWS

CloudFormation tidak mengubah, memodifikasi, atau menyunting informasi apa pun yang Anda sertakan di Metadata bagian ini. Nilai default-nya adalah `false`.

Untuk informasi selengkapnya tentang penggunaan `NoEcho` untuk menutupi informasi sensitif, lihat praktik [Jangan menanamkan kredensial dalam templat Anda](#) terbaik.

Data

Opsional. Pasangan nama-nilai yang ditentukan penyedia sumber daya khusus untuk dikirim dengan respons. Anda dapat mengakses nilai yang disediakan di sini dengan nama di template dengan `Fn::GetAtt`.

Important

Jika pasangan nama-nilai berisi informasi sensitif, Anda harus menggunakan `NoEcho` bidang untuk menutupi output dari sumber daya kustom. Jika tidak, nilai terlihat melalui APIs nilai properti permukaan (seperti `DescribeStackEvents`).

Contoh

```
{
  "Status" : "SUCCESS",
  "RequestId" : "unique id for this create request (copied from request)",
  "LogicalResourceId" : "name of resource in template (copied from request)",
  "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-cd98-11ea-90d5-0a9cd3354c10 (copied from request)",
  "PhysicalResourceId" : "required vendor-defined physical id that is unique for that vendor",
  "Data" : {
    "keyThatCanBeUsedInGetAtt1" : "data for key 1",
    "keyThatCanBeUsedInGetAtt2" : "data for key 2"
  }
}
```

Failed

Ketika membuat permintaan gagal, respons harus dikirim ke bucket S3 dengan bidang berikut:

Status

Harus berupa `FAILED`.

Reason

Menjelaskan alasan respons kegagalan.

RequestId

ID unik untuk permintaan tersebut. Nilai respons ini harus disalin kata demi kata dari permintaan.

LogicalResourceId

Nama template yang dipilih pengembang (ID logis) dari sumber daya kustom dalam template. AWS CloudFormation Nilai respons ini harus disalin kata demi kata dari permintaan.

StackId

Nama Sumber Daya Amazon (ARN) yang mengidentifikasi tumpukan yang berisi sumber daya khusus. Nilai respons ini harus disalin kata demi kata dari permintaan.

PhysicalResourceId

Nilai ini harus menjadi pengenalan yang unik untuk vendor sumber daya khusus, dan dapat berukuran hingga 1 KB. Nilai harus berupa string yang tidak kosong dan harus identik untuk semua tanggapan untuk sumber daya yang sama.

Nilai yang dikembalikan untuk `PhysicalResourceId` dapat mengubah operasi pembaruan sumber daya kustom. Jika nilai yang dikembalikan sama, itu dianggap sebagai pembaruan normal. Jika nilai yang dikembalikan berbeda, AWS CloudFormation mengenali pembaruan sebagai pengganti dan mengirimkan permintaan hapus ke sumber daya lama. Untuk informasi selengkapnya, lihat [AWS::CloudFormation::CustomResource](#).

Contoh

```
{
  "Status" : "FAILED",
  "Reason" : "Required failure reason string",
  "RequestId" : "unique id for this create request (copied from request)",
  "LogicalResourceId" : "name of resource in template (copied from request)",
  "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-cd98-11ea-90d5-0a9cd3354c10 (copied from request)",
  "PhysicalResourceId" : "required vendor-defined physical id that is unique for that vendor"
}
```

Hapus permintaan untuk sumber daya CloudFormation khusus

Saat pengembang template menghapus tumpukan atau menghapus sumber daya khusus dari tumpukan, CloudFormation kirimkan permintaan ke penyedia sumber daya khusus dengan `RequestType` disetel ke `Delete`. Agar berhasil menghapus tumpukan dengan sumber daya kustom, penyedia sumber daya kustom harus berhasil merespons permintaan penghapusan.

Untuk pengenalan sumber daya khusus dan cara kerjanya, lihat [Buat logika penyediaan khusus dengan sumber daya khusus](#).

Permintaan

Hapus permintaan berisi bidang berikut:

`RequestType`

`Delete`.

`RequestId`

ID unik untuk permintaan tersebut.

`ResponseURL`

URL respons mengidentifikasi bucket S3 yang telah ditetapkan sebelumnya yang menerima tanggapan dari penyedia sumber daya khusus. AWS CloudFormation

`ResourceType`

Jenis sumber daya yang dipilih developer templat dari sumber daya kustom di templat CloudFormation. Nama jenis sumber daya kustom dapat memiliki panjang hingga 60 karakter dan dapat mencakup alfanumerik dan karakter berikut: `_@-`.

`LogicalResourceId`

Nama template yang dipilih pengembang (ID logis) dari sumber daya kustom dalam template. AWS CloudFormation

`StackId`

Nama Sumber Daya Amazon (ARN) yang mengidentifikasi tumpukan yang berisi sumber daya khusus.

`PhysicalResourceId`

ID fisik yang ditentukan penyedia sumber daya kustom wajib yang unik untuk penyedia tersebut.

Nilai yang dikembalikan untuk `PhysicalResourceId` dapat mengubah operasi pembaruan sumber daya kustom. Jika nilai yang dikembalikan sama, itu dianggap sebagai pembaruan normal. Jika nilai yang dikembalikan berbeda, AWS CloudFormation mengenali pembaruan sebagai pengganti dan mengirimkan permintaan hapus ke sumber daya lama. Untuk informasi selengkapnya, lihat [AWS::CloudFormation::CustomResource](#).

ResourceProperties

Bidang ini berisi isi `Properties` objek yang dikirim oleh pengembang template. Isinya ditentukan oleh penyedia sumber daya khusus.

Contoh

```
{
  "RequestType" : "Delete",
  "RequestId" : "unique id for this delete request",
  "ResponseURL" : "pre-signed-url-for-delete-response",
  "ResourceType" : "Custom::MyCustomResourceType",
  "LogicalResourceId" : "name of resource in template",
  "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-
cd98-11ea-90d5-0a9cd3354c10",
  "PhysicalResourceId" : "custom resource provider-defined physical id",
  "ResourceProperties" : {
    "key1" : "string",
    "key2" : [ "list" ],
    "key3" : { "key4" : "map" }
  }
}
```

Respons

Berhasil

Ketika menghapus permintaan berhasil, respons harus dikirim ke bucket S3 dengan bidang berikut:

Status

Harus berupa `SUCCESS`.

RequestId

ID unik untuk permintaan tersebut. Nilai respons ini harus disalin kata demi kata dari permintaan.

LogicalResourceId

Nama template yang dipilih pengembang (ID logis) dari sumber daya kustom dalam template. AWS CloudFormation Nilai respons ini harus disalin kata demi kata dari permintaan.

StackId

Nama Sumber Daya Amazon (ARN) yang mengidentifikasi tumpukan yang berisi sumber daya khusus. Nilai respons ini harus disalin kata demi kata dari permintaan.

PhysicalResourceId

Nilai ini harus menjadi pengidentifikasi yang unik untuk vendor sumber daya khusus, dan dapat berukuran hingga 1 KB. Nilai harus berupa string yang tidak kosong dan harus identik untuk semua tanggapan untuk sumber daya yang sama.

Nilai yang dikembalikan untuk PhysicalResourceId dapat mengubah operasi pembaruan sumber daya kustom. Jika nilai yang dikembalikan sama, itu dianggap sebagai pembaruan normal. Jika nilai yang dikembalikan berbeda, AWS CloudFormation mengenali pembaruan sebagai pengganti dan mengirimkan permintaan hapus ke sumber daya lama. Untuk informasi selengkapnya, lihat [AWS::CloudFormation::CustomResource](#).

Contoh

```
{
  "Status" : "SUCCESS",
  "RequestId" : "unique id for this delete request (copied from request)",
  "LogicalResourceId" : "name of resource in template (copied from request)",
  "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-cd98-11ea-90d5-0a9cd3354c10 (copied from request)",
  "PhysicalResourceId" : "custom resource provider-defined physical id"
}
```

Failed

Ketika menghapus permintaan gagal, respons harus dikirim ke bucket S3 dengan bidang berikut:

Status

Harus berupa FAILED.

Reason

Sebab kegagalan.

RequestId

Nilai RequestId disalin dari [hapus permintaan](#).

LogicalResourceId

Nilai LogicalResourceId disalin dari [hapus permintaan](#).

StackId

Nilai StackId disalin dari [hapus permintaan](#).

PhysicalResourceId

ID fisik yang ditentukan penyedia sumber daya khusus wajib yang unik untuk penyedia tersebut.

Contoh

```
{
  "Status" : "FAILED",
  "Reason" : "Required failure reason string",
  "RequestId" : "unique id for this delete request (copied from request)",
  "LogicalResourceId" : "name of resource in template (copied from request)",
  "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-
cd98-11ea-90d5-0a9cd3354c10 (copied from request)",
  "PhysicalResourceId" : "custom resource provider-defined physical id"
}
```

Permintaan pembaruan untuk sumber daya CloudFormation khusus

Ketika pengembang template membuat perubahan pada properti sumber daya kustom dalam template dan memperbarui tumpukan, CloudFormation mengirimkan permintaan ke penyedia sumber daya kustom dengan RequestType set keUpdate. Ini berarti bahwa kode sumber daya kustom Anda tidak harus mendeteksi perubahan sumber daya karena mengetahui bahwa propertinya telah berubah ketika jenis permintaanUpdate.

Untuk pengenalan sumber daya khusus dan cara kerjanya, lihat [Buat logika penyediaan khusus dengan sumber daya khusus](#).

Permintaan

Permintaan Perbarui berisi bidang berikut:

RequestType

Update.

RequestId

ID unik untuk permintaan tersebut.

ResponseURL

URL respons mengidentifikasi bucket S3 yang telah ditetapkan sebelumnya yang menerima tanggapan dari penyedia sumber daya kustom ke. AWS CloudFormation

ResourceType

Jenis sumber daya yang dipilih developer templat dari sumber daya kustom di templat CloudFormation . Nama jenis sumber daya kustom dapat memiliki panjang hingga 60 karakter dan dapat mencakup alfanumerik dan karakter berikut: `_@-` . Anda tidak dapat mengubah jenis selama pembaruan.

LogicalResourceId

Nama template yang dipilih pengembang (ID logis) dari sumber daya kustom dalam template. AWS CloudFormation

StackId

Nama Sumber Daya Amazon (ARN) yang mengidentifikasi tumpukan yang berisi sumber daya khusus.

PhysicalResourceId

ID fisik yang ditentukan penyedia sumber daya kustom wajib yang unik untuk penyedia tersebut.

Nilai yang dikembalikan untuk PhysicalResourceId dapat mengubah operasi pembaruan sumber daya kustom. Jika nilai yang dikembalikan sama, itu dianggap sebagai pembaruan normal. Jika nilai yang dikembalikan berbeda, AWS CloudFormation mengenali pembaruan sebagai pengganti dan mengirimkan permintaan hapus ke sumber daya lama. Untuk informasi selengkapnya, lihat [AWS::CloudFormation::CustomResource](#).

ResourceProperties

Nilai properti sumber daya baru yang dideklarasikan oleh pengembang template dalam CloudFormation template yang diperbarui.

OldResourceProperties

Nilai properti sumber daya yang sebelumnya dideklarasikan oleh pengembang template dalam CloudFormation template.

Contoh

```
{
  "RequestType" : "Update",
  "RequestId" : "unique id for this update request",
  "ResponseURL" : "pre-signed-url-for-update-response",
  "ResourceType" : "Custom::MyCustomResourceType",
  "LogicalResourceId" : "name of resource in template",
  "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-cd98-11ea-90d5-0a9cd3354c10",
  "PhysicalResourceId" : "custom resource provider-defined physical id",
  "ResourceProperties" : {
    "key1" : "new-string",
    "key2" : [ "new-list" ],
    "key3" : { "key4" : "new-map" }
  },
  "OldResourceProperties" : {
    "key1" : "string",
    "key2" : [ "list" ],
    "key3" : { "key4" : "map" }
  }
}
```

Respons

Berhasil

Jika penyedia sumber daya kustom berhasil memperbarui sumber daya, CloudFormation mengharapkan status disetel ke SUCCESS dalam respons.

Status

Harus berupa SUCCESS.

RequestId

ID unik untuk permintaan tersebut. Nilai respons ini harus disalin kata demi kata dari permintaan.

LogicalResourceId

Nama template yang dipilih pengembang (ID logis) dari sumber daya kustom dalam template. AWS CloudFormation Nilai respons ini harus disalin kata demi kata dari permintaan.

StackId

Nama Sumber Daya Amazon (ARN) yang mengidentifikasi tumpukan yang berisi sumber daya khusus. Nilai respons ini harus disalin kata demi kata dari permintaan.

PhysicalResourceId

Nilai ini harus menjadi pengenalan yang unik untuk vendor sumber daya khusus, dan dapat berukuran hingga 1 KB. Nilai harus berupa string yang tidak kosong dan harus identik untuk semua tanggapan untuk sumber daya yang sama.

Nilai yang dikembalikan untuk `PhysicalResourceId` dapat mengubah operasi pembaruan sumber daya kustom. Jika nilai yang dikembalikan sama, itu dianggap sebagai pembaruan normal. Jika nilai yang dikembalikan berbeda, AWS CloudFormation mengenali pembaruan sebagai pengganti dan mengirimkan permintaan hapus ke sumber daya lama. Untuk informasi selengkapnya, lihat [AWS::CloudFormation::CustomResource](#).

NoEcho

Opsional. Menunjukkan apakah akan menutupi output dari sumber daya kustom saat diambil dengan menggunakan `Fn::GetAtt` fungsi. Jika disetel ke `true`, semua nilai yang dikembalikan ditutupi dengan tanda bintang (`*****`), kecuali yang disimpan di bagian *Metadata* template. AWS CloudFormation tidak mengubah, memodifikasi, atau menyunting informasi apa pun yang Anda sertakan di *Metadata* bagian ini. Nilai default-nya adalah `false`.

Untuk informasi selengkapnya tentang penggunaan `NoEcho` untuk menutupi informasi sensitif, lihat praktik [Jangan menanamkan kredensial dalam templat Anda](#) terbaik.

Data

Opsional. Pasangan nama-nilai yang ditentukan penyedia sumber daya khusus untuk dikirim dengan respons. Anda dapat mengakses nilai yang disediakan di sini dengan nama di template dengan `Fn::GetAtt`.

⚠ Important

Jika pasangan nama-nilai berisi informasi sensitif, Anda harus menggunakan NoEcho bidang untuk menutupi output dari sumber daya kustom. Jika tidak, nilai terlihat melalui APIs nilai properti permukaan (seperti `DescribeStackEvents`).

Contoh

```
{
  "Status" : "SUCCESS",
  "RequestId" : "unique id for this update request (copied from request)",
  "LogicalResourceId" : "name of resource in template (copied from request)",
  "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-cd98-11ea-90d5-0a9cd3354c10 (copied from request)",
  "PhysicalResourceId" : "custom resource provider-defined physical id",
  "Data" : {
    "keyThatCanBeUsedInGetAtt1" : "data for key 1",
    "keyThatCanBeUsedInGetAtt2" : "data for key 2"
  }
}
```

Failed

Jika sumber daya tidak dapat diperbarui dengan kumpulan properti baru, CloudFormation mengharapkan status `FAILED`, bersama dengan alasan kegagalan dalam respons.

Status

Harus berupa `FAILED`.

Reason

Menjelaskan alasan respons kegagalan.

RequestId

ID unik untuk permintaan tersebut. Nilai respons ini harus disalin kata demi kata dari permintaan.

LogicalResourceId

Nama template yang dipilih pengembang (ID logis) dari sumber daya kustom dalam template. AWS CloudFormation Nilai respons ini harus disalin kata demi kata dari permintaan.

StackId

Nama Sumber Daya Amazon (ARN) yang mengidentifikasi tumpukan yang berisi sumber daya khusus. Nilai respons ini harus disalin kata demi kata dari permintaan.

PhysicalResourceId

Nilai ini harus menjadi pengenalan yang unik untuk vendor sumber daya khusus, dan dapat berukuran hingga 1 KB. Nilai harus berupa string yang tidak kosong dan harus identik untuk semua tanggapan untuk sumber daya yang sama.

Nilai yang dikembalikan untuk `PhysicalResourceId` dapat mengubah operasi pembaruan sumber daya kustom. Jika nilai yang dikembalikan sama, itu dianggap sebagai pembaruan normal. Jika nilai yang dikembalikan berbeda, AWS CloudFormation mengenali pembaruan sebagai pengganti dan mengirimkan permintaan hapus ke sumber daya lama. Untuk informasi selengkapnya, lihat [AWS::CloudFormation::CustomResource](#).

Contoh

```
{
  "Status" : "FAILED",
  "Reason" : "Required failure reason string",
  "RequestId" : "unique id for this update request (copied from request)",
  "LogicalResourceId" : "name of resource in template (copied from request)",
  "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-cd98-11ea-90d5-0a9cd3354c10 (copied from request)",
  "PhysicalResourceId" : "custom resource provider-defined physical id"
}
```

Lakukan pemrosesan kustom pada CloudFormation template dengan makro template

Dengan makro, Anda dapat melakukan pemrosesan khusus pada templat, dari tindakan sederhana seperti find-and-replace operasi hingga transformasi ekstensif seluruh templat.

Untuk mendapatkan gambaran tentang luasnya kemungkinan, pertimbangkan `AWS::Include` dan `AWS::Serverless` transformasi, yang merupakan makro yang dihosting oleh: CloudFormation

- [AWS::Includetransform](#) memungkinkan Anda untuk memasukkan cuplikan template boilerplate ke dalam template Anda.

- [AWS::Serverlesstransform](#) mengambil seluruh template yang ditulis dalam sintaks AWS Serverless Application Model (AWS SAM) dan mengubah dan memperluasnya menjadi template yang sesuai. CloudFormation Untuk informasi selengkapnya tentang aplikasi tanpa server dan AWS SAM, lihat Panduan [AWS Serverless Application Model Pengembang](#).

Topik

- [Penagihan](#)
- [Contoh makro](#)
- [Sumber daya terkait](#)
- [Ikhtisar CloudFormation makro](#)
- [Buat definisi CloudFormation makro](#)
- [Contoh makro pengganti string sederhana](#)
- [Memecahkan masalah template yang diproses](#)

Penagihan

Saat makro berjalan, pemilik fungsi Lambda ditagih untuk biaya apa pun yang terkait dengan eksekusi fungsi tersebut.

`AWS::Include` dan `AWS::Serverless` transformasi adalah makro yang dihosting oleh CloudFormation. Tidak ada biaya untuk menggunakannya.

Contoh makro

Selain contoh di bagian ini, Anda dapat menemukan contoh makro, termasuk kode sumber dan templat, di [GitHub repositori](#) kami. Contoh-contoh ini disediakan 'sebagaimana adanya' untuk tujuan instruksional.

Sumber daya terkait

- [AWS::CloudFormation::Macro](#)
- [CloudFormation Transform bagian template](#)
- [Fn::Transform](#)
- [AWS::Serverless mengubah](#)
- [AWS::Include mengubah](#)

Ikhtisar CloudFormation makro

Ada dua langkah utama untuk memproses templat menggunakan makro: membuat makro itu sendiri, lalu menggunakan makro untuk melakukan pemrosesan pada templat Anda.

Untuk membuat definisi makro, Anda harus membuat yang berikut:

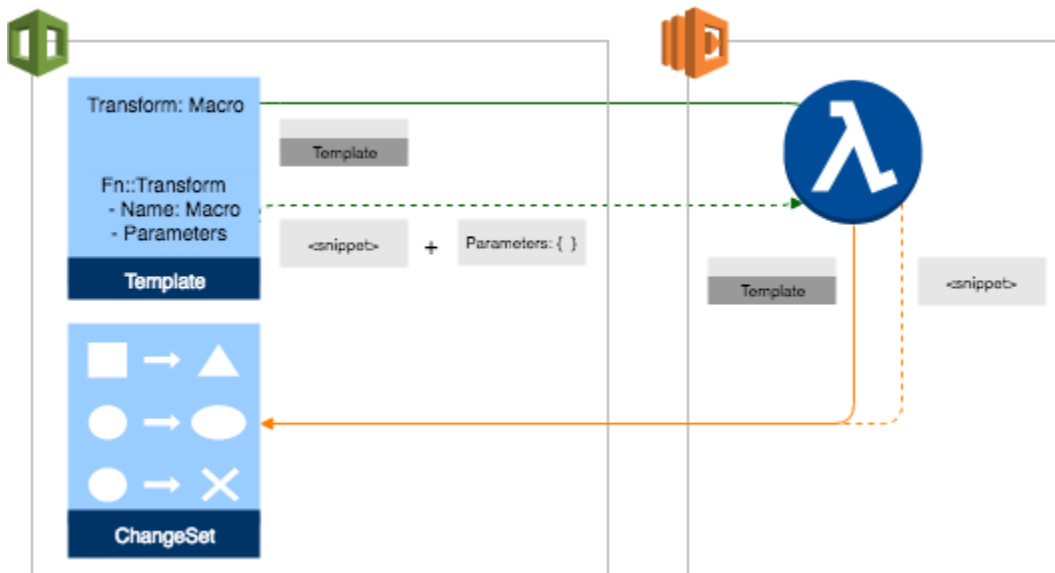
- Fungsi Lambda untuk melakukan pemrosesan template. Fungsi Lambda ini menerima cuplikan atau seluruh templat, dan parameter tambahan apa pun yang Anda tentukan. Fungsi ini mengembalikan snippet templat yang diproses atau seluruh templat sebagai respons.
- Sumber daya tipe [AWS::CloudFormation::Macro](#), yang memungkinkan pengguna untuk memanggil fungsi Lambda dari dalam CloudFormation templat. Sumber daya ini menentukan ARN dari fungsi Lambda untuk memanggil makro ini, dan properti opsional tambahan untuk membantu debugging. Untuk membuat sumber daya ini di dalam akun, buat templat yang menyertakan `AWS::CloudFormation::Macro` sumber daya, lalu buat tumpukan atau kumpulan tumpukan dengan izin yang dikelola sendiri dari templat. AWS CloudFormation StackSet saat ini tidak mendukung pembuatan atau pembaruan kumpulan tumpukan dengan izin yang dikelola layanan dari templat yang mereferensikan makro.

Untuk menggunakan makro, referensikan makro dalam templat Anda:

- Untuk memproses bagian, atau bagian, dari template, referensi makro dalam `Fn::Transform` fungsi yang terletak relatif terhadap konten template yang ingin Anda ubah. Saat menggunakan `Fn::Transform`, Anda juga dapat melewati parameter tertentu yang diperlukan.
- Untuk memproses seluruh template, referensikan makro di bagian [Transform](#) dari templat.

Selanjutnya, Anda biasanya membuat set perubahan dan kemudian menjalankannya. (Memproses makro dapat menambahkan beberapa sumber daya yang mungkin tidak Anda sadari. Untuk memastikan bahwa Anda mengetahui semua perubahan yang diperkenalkan oleh makro, kami sangat menyarankan agar Anda menggunakan set perubahan.) CloudFormation meneruskan konten template yang ditentukan, bersama dengan parameter tambahan yang ditentukan, ke fungsi Lambda yang ditentukan dalam sumber daya makro. Fungsi Lambda mengembalikan konten template yang diproses, baik itu cuplikan atau seluruh template.

Setelah semua makro dalam template dipanggil, CloudFormation menghasilkan set perubahan yang menyertakan konten template yang diproses. Setelah Anda meninjau set perubahan, jalankan untuk menerapkan perubahan.



Cara membuat tumpukan secara langsung

Untuk membuat atau memperbarui tumpukan menggunakan templat yang mereferensikan makro, Anda biasanya membuat set perubahan, lalu menjalankannya. Set perubahan menjelaskan tindakan yang akan diambil CloudFormation berdasarkan templat yang diproses. Memproses makro dapat menambahkan beberapa sumber daya yang mungkin tidak Anda sadari. Untuk memastikan bahwa Anda mengetahui semua perubahan yang diperkenalkan oleh makro, kami sangat menyarankan Anda menggunakan set perubahan. Setelah Anda meninjau set perubahan, Anda dapat menjalankannya untuk benar-benar menerapkan perubahan.

Makro dapat menambahkan sumber daya IAM ke template Anda. Untuk sumber daya ini, CloudFormation mengharuskan Anda untuk [mengakui kemampuan mereka](#). Karena tidak CloudFormation dapat mengetahui sumber daya mana yang ditambahkan sebelum memproses template Anda, Anda mungkin perlu mengakui kemampuan IAM saat membuat set perubahan, tergantung pada apakah makro yang direferensikan berisi sumber daya IAM. Dengan begitu, ketika Anda menjalankan set perubahan, CloudFormation memiliki kemampuan yang diperlukan untuk membuat sumber daya IAM.

Untuk membuat atau memperbarui tumpukan langsung dari template yang diproses tanpa terlebih dahulu meninjau perubahan yang diusulkan dalam set perubahan, tentukan `CAPABILITY_AUTO_EXPAND` kemampuan selama `UpdateStack` permintaan `CreateStack` atau permintaan. Anda hanya boleh membuat tumpukan langsung dari templat tumpukan yang berisi makro jika Anda tahu apa yang dilakukan pemrosesan makro. Anda tidak dapat menggunakan set perubahan dengan makro set tumpukan; Anda harus memperbarui set tumpukan Anda secara langsung.

Untuk informasi selengkapnya, lihat [CreateStack](#) atau [UpdateStack](#) di Referensi AWS CloudFormation API.

 Important

Jika templat set tumpukan Anda mereferensikan satu atau beberapa makro, Anda harus membuat set tumpukan langsung dari templat yang diproses, tanpa terlebih dahulu meninjau perubahan yang dihasilkan dalam set perubahan. Memproses makro dapat menambahkan beberapa sumber daya yang mungkin tidak Anda sadari. Sebelum Anda membuat atau memperbarui set tumpukan dari templat yang mereferensikan makro secara langsung, pastikan Anda mengetahui apa yang dilakukan pemrosesan makro.

Untuk mengurangi jumlah langkah untuk meluncurkan tumpukan dari template yang mereferensikan makro, Anda dapat menggunakan perintah `package and deploy` AWS CLI . Untuk informasi selengkapnya, lihat [Unggah artefak lokal ke bucket S3 dengan AWS CLI](#) dan [Buat tumpukan yang mencakup transformasi](#).

Pertimbangan

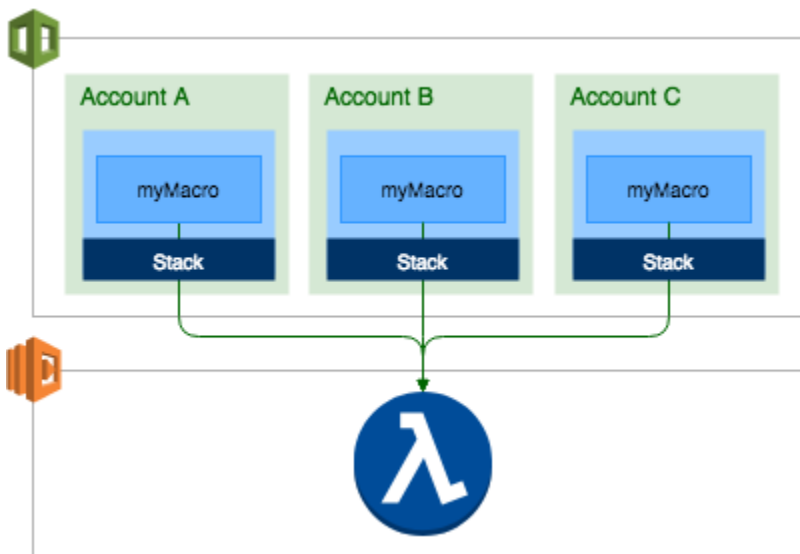
Saat bekerja dengan makro, ingatlah catatan dan batasan berikut:

- Makro hanya didukung di Wilayah AWS tempat Lambda tersedia. Untuk daftar Wilayah di mana Lambda tersedia, lihat [AWS Lambda titik akhir](#) dan kuota.
- Setiap snippet templat yang diproses harus berupa JSON yang valid.
- Setiap snippet templat yang diproses harus lulus pemeriksaan validasi untuk operasi membuat tumpukan, memperbarui tumpukan, membuat set tumpukan, atau memperbarui set tumpukan.
- CloudFormation menyelesaikan makro terlebih dahulu, dan kemudian memproses template. Templat yang dihasilkan harus berupa JSON yang valid dan tidak boleh melebihi batas ukuran templat.
- Karena urutan CloudFormation proses elemen dalam template, makro tidak dapat menyertakan modul dalam konten template yang diproses yang dikembalikan. CloudFormation Untuk informasi selengkapnya, lihat [Urutan evaluasi makro](#).
- Saat menggunakan fitur update rollback, CloudFormation gunakan salinan template asli. Itu memutar kembali ke templat asli bahkan jika snippet yang disertakan diubah.
- Menyertakan makro dalam makro tidak berfungsi karena kami tidak memproses makro secara rekursif.

- Fungsi intrinsik `Fn::ImportValue` saat ini tidak didukung dalam makro.
- Fungsi intrinsik yang termasuk dalam templat dievaluasi setelah makro apapun. Oleh karena itu, konten template yang diproses yang dikembalikan makro Anda dapat menyertakan panggilan ke fungsi intrinsik, dan mereka dievaluasi seperti biasa.
- `StackSets` saat ini tidak mendukung pembuatan atau pembaruan kumpulan tumpukan dengan izin yang dikelola layanan dari templat yang mereferensikan makro. CloudFormation

Cakupan akun makro dan izin

Anda dapat menggunakan makro hanya di akun tempat mereka dibuat sebagai sumber daya. Nama makro harus unik dalam akun tertentu. Namun, Anda dapat membuat fungsionalitas yang sama tersedia di beberapa akun dengan mengaktifkan akses lintas akun pada fungsi Lambda yang mendasarinya, dan kemudian membuat definisi makro yang mereferensikan fungsi tersebut di beberapa akun. Dalam contoh di bawah ini, tiga akun berisi definisi makro yang masing-masing menunjuk ke fungsi Lambda yang sama.



Untuk membuat definisi makro, pengguna harus memiliki izin untuk membuat tumpukan atau tumpukan yang ditetapkan dalam akun yang ditentukan.

CloudFormation Agar berhasil menjalankan makro yang disertakan dalam template, pengguna harus memiliki `Invoke` izin untuk fungsi Lambda yang mendasarinya. Untuk mencegah potensi eskalasi izin, CloudFormation meniru pengguna saat menjalankan makro.

Untuk informasi selengkapnya, lihat [Mengelola izin AWS Lambda di Panduan AWS Lambda Pengembang](#) dan [Tindakan, sumber daya, dan kunci kondisi AWS Lambda](#) di Referensi Otorisasi Layanan.

Buat definisi CloudFormation makro

Saat Anda membuat definisi makro, definisi makro membuat fungsi Lambda yang mendasarinya tersedia di akun yang ditentukan sehingga CloudFormation memanggilnya untuk memproses templat.

Pemetaan acara

Ketika CloudFormation memanggil fungsi Lambda makro, ia mengirimkan permintaan dalam format JSON dengan struktur berikut:

```
{
  "region" : "us-east-1",
  "accountId" : "$ACCOUNT_ID",
  "fragment" : { ... },
  "transformId" : "$TRANSFORM_ID",
  "params" : { ... },
  "requestId" : "$REQUEST_ID",
  "templateParameterValues" : { ... }
}
```

- **region**

Wilayah tempat makro berada.

- **accountId**

ID akun tempat makro menjalankan fungsi Lambda.

- **fragment**

Konten templat yang tersedia untuk pemrosesan kustom, dalam format JSON.

- Untuk makro yang termasuk dalam bagian templat `Transform`, ini adalah seluruh templat kecuali untuk bagian `Transform`.
- Untuk makro yang disertakan dalam panggilan fungsi intrinsik `Fn::Transform`, ini mencakup semua node saudara (dan turunannya) berdasarkan lokasi fungsi intrinsik dalam templat kecuali untuk fungsi `Fn::Transform` tersebut. Untuk informasi selengkapnya, lihat [Ruang lingkup template makro](#).

- **transformId**

Nama makro yang memanggil fungsi ini.

- **params**

Untuk panggilan Fn::Transform fungsi, parameter apa pun yang ditentukan untuk fungsi tersebut. CloudFormation tidak mengevaluasi parameter ini sebelum meneruskannya ke fungsi.

Untuk makro yang termasuk dalam bagian templat Transform, bagian ini kosong.

- `requestId`

ID permintaan yang memanggil fungsi ini.

- `templateParameterValues`

Parameter apa pun yang ditentukan di [Parameters](#) bagian templat. CloudFormation mengevaluasi parameter ini sebelum meneruskannya ke fungsi.

Format respons

CloudFormation mengharapkan fungsi Lambda mengembalikan respons dalam format JSON berikut:

```
{
  "requestId" : "$REQUEST_ID",
  "status" : "$STATUS",
  "fragment" : { ... },
  "errorMessage": "optional error message for failures"
}
```

- `requestId`

ID permintaan yang memanggil fungsi ini. Ini harus sesuai dengan ID permintaan yang disediakan oleh CloudFormation saat menjalankan fungsi.

- `status`

Status permintaan (peka huruf besar kecil). Harus diatur kesuccess. CloudFormation memperlakukan setiap respon lain sebagai kegagalan.

- `fragment`

Konten template yang diproses CloudFormation untuk disertakan dalam template yang diproses, termasuk saudara kandung. CloudFormation menggantikan konten template yang diteruskan ke fungsi Lambda dengan fragmen template yang diterimanya dalam respons Lambda.

Konten templat yang diproses harus berupa JSON yang valid, dan penyertaannya dalam templat yang diproses harus menghasilkan templat yang valid.

Jika fungsi Anda tidak benar-benar mengubah konten templat yang diteruskan CloudFormation ke sana, tetapi Anda masih perlu menyertakan konten itu dalam templat yang diproses, fungsi Anda perlu mengembalikan konten templat itu ke CloudFormation dalam responsnya.

- `errorMessage`

Pesan kesalahan yang menjelaskan mengapa transformasi gagal. CloudFormation menampilkan pesan kesalahan ini di panel Peristiwa di halaman Detail tumpukan untuk tumpukan Anda.

Misalnya:

```
Error creating change set: Transform
    Akun AWS account
    number::macro name failed with:
    error message string.
```

Buat definisi makro

Untuk membuat definisi CloudFormation makro

1. [Bangun fungsi Lambda](#) yang akan menangani pemrosesan konten template. Hal ini dapat memproses setiap bagian dari template, sampai ke seluruh template.
2. Buat CloudFormation template yang berisi jenis `AWS::CloudFormation::Macro` sumber daya dan tentukan `Name` dan `FunctionName` properti. `FunctionName` properti harus berisi ARN dari fungsi Lambda untuk dipanggil saat menjalankan makro. CloudFormation
3. (Opsional) Untuk membantu dalam debugging, Anda juga dapat menentukan `LogGroupName` dan `LogRoleArn` properti saat membuat jenis `AWS::CloudFormation::Macro` sumber daya untuk makro Anda. Properti ini memungkinkan Anda menentukan grup CloudWatch log Log yang CloudFormation mengirimkan informasi pencatatan kesalahan saat menjalankan fungsi Lambda dasar makro, dan peran tersebut CloudFormation harus diambil saat mengirim entri log ke log tersebut.
4. [Buat tumpukan](#) menggunakan template dengan makro di akun yang ingin Anda gunakan. Atau, [buat kumpulan tumpukan dengan izin yang dikelola sendiri](#) menggunakan templat dengan makro di akun administrator, lalu buat instance tumpukan di akun target.

5. Setelah CloudFormation berhasil membuat tumpukan yang berisi definisi makro, makro tersedia untuk digunakan dalam akun tersebut. Anda menggunakan makro dengan mereferensikannya di templat, di lokasi yang sesuai yang relevan dengan konten templat yang ingin Anda proses.

Ruang lingkup template makro

Makro yang direferensikan di bagian Transform dari templat dapat memproses seluruh konten templat tersebut.

Makro yang direferensikan dalam suatu fungsi Fn::Transform dapat memproses konten salah satu elemen saudara (termasuk anak-anak) dari fungsi Fn::Transform tersebut dalam templat.

Misalnya, pada contoh templat di bawah ini, AWS::Include dapat memproses properti MyBucket, berdasarkan lokasi fungsi Fn::Transform yang memuatnya. MyMacro dapat memproses konten seluruh templat karena penyertaannya di bagian Transform.

```
# 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
```

Urutan evaluasi makro

Anda dapat mereferensikan beberapa makro dalam templat tertentu, termasuk transformasi yang dihosting oleh CloudFormation, seperti [AWS::Include](#) dan [AWS::Serverless](#).

Macro dievaluasi secara berurutan, berdasarkan lokasi mereka di templat, dari yang paling dalam hingga yang paling umum. Makro di lokasi yang sama dalam template dievaluasi secara serial berdasarkan urutan daftar.

Transformasi seperti `AWS::Include` dan `AWS::Transform` diperlakukan sama seperti makro lainnya dalam hal urutan tindakan dan cakupan.

Misalnya, dalam contoh templat di bawah ini, CloudFormation evaluasi `PolicyAdder` makro terlebih dahulu, karena makro ini adalah makro yang paling dalam di template. CloudFormation kemudian mengevaluasi `MyMacro` sebelum mengevaluasi `AWS::Serverless` karena tercantum sebelumnya `AWS::Serverless` di bagian. `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
```

Contoh makro pengganti string sederhana

Contoh berikut memandu Anda melalui proses menggunakan makro, dari mendefinisikan makro dalam template, hingga membuat fungsi Lambda untuk makro, dan kemudian menggunakan makro dalam template.

Dalam contoh ini, kita membuat makro sederhana yang menyisipkan string yang ditentukan di tempat konten target yang ditentukan dalam templat yang diproses. Kemudian kita akan menggunakannya untuk menyisipkan `WaitHandleCondition` kosong di lokasi yang ditentukan dalam templat yang diproses.

Membuat makro

Sebelum menggunakan makro, pertama-tama kita harus menyelesaikan dua hal: membuat fungsi Lambda yang melakukan pemrosesan template yang diinginkan, dan kemudian membuat fungsi Lambda itu tersedia dengan membuat definisi CloudFormation makro.

Contoh template berikut berisi definisi untuk contoh makro kita. Untuk membuat makro tersedia secara spesifik Akun AWS, buat tumpukan dari template. Definisi makro menentukan nama makro, deskripsi singkat, dan referensi ARN dari fungsi Lambda yang dipanggil ketika makro ini CloudFormation digunakan dalam template. (Kami belum menyertakan `LogRoleARN` properti `LogGroupName` atau untuk pencatatan kesalahan.)

Dalam contoh ini, asumsikan bahwa tumpukan yang dibuat dari template ini diberi nama `JavaMacroFunc`. Karena `Name` properti makro diatur ke nama tumpukan, makro yang dihasilkan dinamai `JavaMacroFunc` juga.

```
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'
```

Menggunakan makro

Untuk menggunakan makro kami, kami memasukkannya ke dalam template menggunakan fungsi `Fn::Transform` intrinsik.

Ketika kita membuat tumpukan menggunakan template di bawah ini, CloudFormation memanggil contoh makro kita. Fungsi Lambda yang mendasari menggantikan satu string tertentu dengan string lain yang ditentukan. Dalam hal ini, hasilnya kosong `AWS::CloudFormation::WaitConditionHandle` dimasukkan ke dalam template yang diproses.

```
Parameters:
```

```
ExampleParameter:
  Type: String
  Default: 'SampleMacro'

Resources:
  2a:
    Fn::Transform:
      Name: "JavaMacroFunc"
      Parameters:
        replacement: 'AWS::CloudFormation::WaitConditionHandle'
        target: '$$REPLACEMENT$$'
      Type: '$$REPLACEMENT$$'
```

- Makro untuk memanggil ditentukan sebagai `JavaMacroFunc`, yang berasal dari contoh definisi makro sebelumnya.
- Makro melewati dua parameter, `target` dan `replacement`, yang mewakili string target dan nilai penggantian yang diinginkan.
- Makro dapat beroperasi pada konten simpul `Type` karena merupakan `Type` saudara dari fungsi `Fn::Transform` yang mereferensikan makro.
- Hasilnya `AWS::CloudFormation::WaitConditionHandle` dinamai `2a`.
- Template juga berisi parameter template, `ExampleParameter`, yang juga dapat diakses oleh makro (tetapi tidak digunakan dalam hal ini).

Data masukan Lambda

Saat CloudFormation memproses contoh template kami selama pembuatan tumpukan, ia meneruskan pemetaan peristiwa berikut ke fungsi Lambda yang direferensikan dalam `JavaMacroFunc` definisi makro.

- `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 berisi JSON yang mewakili fragmen templat yang dapat diproses makro. Fragmen ini terdiri dari saudara dari panggilan fungsi `Fn::Transform`, tetapi bukan fungsi panggilan itu sendiri. Juga, `params` berisi JSON yang mewakili parameter makro. Dalam hal ini, penggantian dan target. Demikian pula, `templateParameterValues` berisi JSON yang mewakili parameter yang ditentukan untuk templat secara keseluruhan.

Kode fungsi Lambda

Berikut ini adalah kode aktual untuk fungsi Lambda yang mendasari `JavaMacroFunc` contoh makro. Ini mengulangi fragmen templat yang disertakan dalam respons (baik dalam format string, daftar, maupun peta), mencari string target yang ditentukan. Jika menemukan string target yang ditentukan, fungsi Lambda menggantikan string target dengan string pengganti yang ditentukan. Jika tidak, fungsi membiarkan fragmen template tidak berubah. Kemudian, fungsi mengembalikan peta properti yang diharapkan, dibahas secara rinci di bawah ini, ke 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;
}
}

```

Respon fungsi Lambda

Berikut ini adalah pemetaan yang fungsi Lambda kembali CloudFormation untuk diproses.

- `requestId` : 5dba79b5-f117-4de0-9ce4-d40363bfb6ab
- `status` : SUCCESS
- `fragment` :

```

{
  "Type": "AWS::CloudFormation::WaitConditionHandle"
}

```

`requestId` Pencocokan yang dikirim dari CloudFormation, dan `status` nilai SUCCESS menunjukkan bahwa fungsi Lambda berhasil memproses fragmen template yang disertakan dalam permintaan. Dalam respons ini, `fragment` berisi JSON mewakili konten yang akan dimasukkan ke dalam templat yang diproses sebagai pengganti snippet templat asli.

Template yang diproses yang dihasilkan

Setelah CloudFormation menerima respons yang berhasil dari fungsi Lambda, ia menyisipkan fragmen template yang dikembalikan ke dalam template yang diproses.

Di bawah ini adalah templat yang diproses yang dihasilkan untuk contoh kita. Panggilan fungsi `Fn::Transform` intrinsik yang mereferensikan `JavaMacroFunc` makro tidak lagi disertakan. Fragmen template yang dikembalikan oleh fungsi Lambda disertakan di lokasi yang sesuai, dengan hasil bahwa `"Type": "$$REPLACEMENT$$"` konten telah diganti. `"Type": "AWS::CloudFormation::WaitConditionHandle"`


```
{
  "Parameters": {
    "ExampleParameter": {
      "Default": "SampleMacro",
      "Type": "String"
    }
  },
  "Resources": {
    "2a": {
      "Type": "AWS::CloudFormation::WaitConditionHandle"
    }
  }
}
```

Memecahkan masalah template yang diproses

Saat menggunakan makro, templat yang diproses dapat ditemukan di CloudFormation konsol.

Tahap template menunjukkan status pemrosesannya:

- **Original:** Templat yang awalnya dikirimkan pengguna untuk membuat atau memperbarui tumpukan atau set tumpukan.
- **Processed:** Template yang CloudFormation digunakan untuk membuat atau memperbarui tumpukan atau set tumpukan setelah memproses makro yang direferensikan. Templat yang diproses diformat sebagai JSON, bahkan jika templat asli diformat sebagai YAML.

Untuk pemecahan masalah, gunakan templat yang diproses. Jika template tidak mereferensikan makro, templat asli dan yang diproses akan serupa.

Untuk informasi selengkapnya, lihat [Lihat informasi tumpukan dari CloudFormation konsol](#).

Untuk menggunakan AWS CLI untuk mendapatkan template yang diproses, gunakan [get-template](#) perintah.

Batasan ukuran

Ukuran maksimum untuk template tumpukan yang diproses adalah 51.200 byte ketika diteruskan langsung ke, atau `ValidateTemplate` permintaan `CreateStackUpdateStack`, atau 1 MB saat diteruskan sebagai objek S3 menggunakan URL template Amazon S3. Namun, selama pemrosesan CloudFormation memperbarui status sementara template karena secara serial memproses makro yang terkandung dalam template. Karena itu, ukuran template selama pemrosesan dapat

sementara melebihi ukuran yang diizinkan dari templat yang diproses sepenuhnya. CloudFormation memungkinkan beberapa buffer untuk template dalam proses ini. Namun, Anda harus mendesain templat dan makro dengan mengingat ukuran maksimum yang diizinkan untuk templat tumpukan yang diproses.

Jika CloudFormation mengembalikan `Transformation data limit exceeded` kesalahan saat memproses template Anda, template Anda telah melebihi ukuran template maksimum yang CloudFormation memungkinkan selama pemrosesan.

Untuk mengatasi masalah ini, pertimbangkan untuk melakukan hal berikut:

- Susun ulang templat Anda menjadi beberapa templat untuk menghindari melebihi ukuran maksimum untuk templat dalam proses. Sebagai contoh:
 - Gunakan templat tumpukan nested untuk merangkum bagian dari templat. Untuk informasi selengkapnya, lihat [Pisahkan template menjadi potongan-potongan yang dapat digunakan kembali menggunakan tumpukan bersarang](#).
 - Buat beberapa tumpukan dan gunakan referensi lintas tumpukan untuk saling bertukar informasi. Untuk informasi selengkapnya, lihat [Lihat output sumber daya di tumpukan lain CloudFormation](#).
- Kurangi ukuran fragmen templat yang dikembalikan oleh makro tertentu. CloudFormation tidak merusak konten fragmen yang dikembalikan oleh makro.

Pisahkan template menjadi potongan-potongan yang dapat digunakan kembali menggunakan tumpukan bersarang

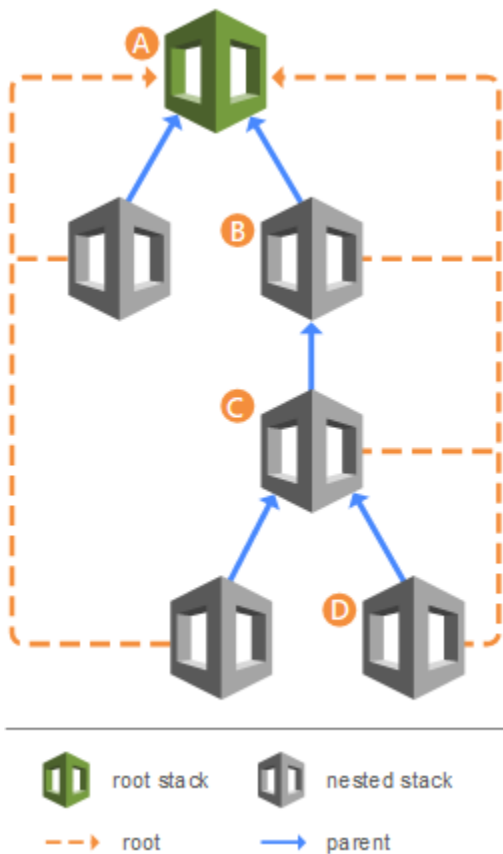
Seiring pertumbuhan infrastruktur Anda, Anda mungkin menemukan diri Anda berulang kali membuat konfigurasi sumber daya yang identik di beberapa templat. Untuk menghindari redundansi ini, Anda dapat memisahkan konfigurasi umum ini menjadi template khusus. Kemudian, Anda dapat menggunakan `AWS::CloudFormation::Stack` sumber daya di templat lain untuk mereferensikan templat khusus ini, membuat tumpukan bersarang.

Misalnya, Anda memiliki konfigurasi penyeimbang beban yang Anda gunakan untuk sebagian besar tumpukan Anda. Alih-alih menyalin dan menempelkan konfigurasi yang sama ke dalam templat, Anda dapat membuat templat khusus untuk penyeimbang beban. Kemudian, Anda dapat mereferensikan template ini dari dalam template lain yang memerlukan konfigurasi penyeimbang beban yang sama.

Tumpukan bersarang sendiri dapat berisi tumpukan bersarang lainnya, menghasilkan hierarki tumpukan, seperti yang ditunjukkan pada diagram di bawah ini. Tumpukan root adalah tumpukan

tingkat atas yang menjadi milik semua tumpukan bersarang. Setiap tumpukan bersarang memiliki tumpukan induk langsung. Untuk tingkat pertama tumpukan bersarang, tumpukan root juga merupakan tumpukan induk.

- Tumpukan A adalah tumpukan akar untuk seluruh tumpukan nest dalam hierarki.
- Untuk tumpukan B, tumpukan A adalah tumpukan induk sekaligus tumpukan akar.
- Untuk tumpukan D, tumpukan C adalah tumpukan induk; sedangkan untuk tumpukan C, tumpukan B adalah tumpukan induk.



Topik

- [Sebelum dan sesudah contoh pemisahan template](#)
- [Contoh arsitektur tumpukan bersarang](#)
- [Melakukan operasi tumpukan pada tumpukan bersarang](#)
- [Informasi terkait](#)

Sebelum dan sesudah contoh pemisahan template

Contoh ini menunjukkan bagaimana Anda dapat mengambil satu CloudFormation template besar dan mengaturnya kembali menjadi desain yang lebih terstruktur dan dapat digunakan kembali menggunakan templat bersarang. Awalnya, template “Sebelum tumpukan bersarang” menunjukkan semua sumber daya yang ditentukan dalam satu file. Ini bisa menjadi berantakan dan sulit dikelola seiring bertambahnya jumlah sumber daya. Template “After nesting stacks” membagi sumber daya menjadi template yang lebih kecil dan terpisah. Setiap tumpukan bersarang menangani serangkaian sumber daya terkait tertentu, membuat struktur keseluruhan lebih terorganisir dan lebih mudah dirawat.

Sebelum tumpukan bersarang	Setelah tumpukan bersarang
<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/ <i>amzn-s3-demo-bucket</i> /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/ <i>amzn-s3-demo-bucket</i> /second-nested-stack.yaml Parameters: # Pass parameters to the nested stack if needed Environment: Testing DependsOn: MyFirstNestedStack </pre>

Contoh arsitektur tumpukan bersarang

Bagian ini menunjukkan arsitektur tumpukan bersarang yang terdiri dari tumpukan tingkat atas yang mereferensikan tumpukan bersarang. Tumpukan bersarang menerapkan fungsi Lambda Node.js, menerima nilai parameter dari tumpukan tingkat atas, dan mengembalikan output yang diekspos melalui tumpukan tingkat atas.

Topik

- [Langkah 1: Buat template untuk tumpukan bersarang di sistem lokal Anda](#)
- [Langkah 2: Buat template untuk tumpukan tingkat atas pada sistem lokal Anda](#)
- [Langkah 3: Package dan deploy template](#)

Langkah 1: Buat template untuk tumpukan bersarang di sistem lokal Anda

Contoh berikut menunjukkan format template stack bersarang.

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

```

Langkah 2: Buat template untuk tumpukan tingkat atas pada sistem lokal Anda

Contoh berikut menunjukkan format template tumpukan tingkat atas dan

[AWS::CloudFormation::Stack](#) sumber daya yang mereferensikan tumpukan yang Anda buat pada langkah sebelumnya.

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

```

Langkah 3: Package dan deploy template

Note

Saat bekerja dengan templat secara lokal, AWS CLI package perintah dapat membantu Anda menyiapkan templat untuk penerapan. Ini secara otomatis menangani unggahan artefak lokal ke Amazon S3 (TemplateURL termasuk) dan menghasilkan file template baru dengan referensi yang diperbarui ke lokasi S3 ini. Untuk informasi selengkapnya, lihat [Unggah artefak lokal ke bucket S3 dengan AWS CLI](#).

Selanjutnya, Anda dapat menggunakan [package](#) perintah untuk mengunggah template bersarang ke bucket Amazon S3.

```
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
```

Perintah menghasilkan template baru di jalur yang ditentukan oleh `--output-template-file`. Ini menggantikan TemplateURL referensi dengan lokasi Amazon S3, seperti yang ditunjukkan di bawah ini.

Template yang dihasilkan

```
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
```

Setelah Anda menjalankan `package` perintah, Anda dapat menggunakan template yang diproses menggunakan `deploy` perintah. Untuk tumpukan bersarang yang berisi sumber daya IAM, Anda harus mengakui kemampuan IAM dengan menyertakan opsi. `--capabilities`

```
aws cloudformation deploy \  
  --template-file packaged-template.yaml \  
  --stack-name stack-name \  
  --capabilities CAPABILITY_NAMED_IAM
```

Melakukan operasi tumpukan pada tumpukan bersarang

Saat bekerja dengan tumpukan bersarang, Anda harus menanganinya dengan hati-hati selama operasi. Operasi tumpukan tertentu, seperti pembaruan tumpukan, harus dimulai dari tumpukan root daripada dilakukan langsung pada tumpukan bersarang. Saat Anda memperbarui tumpukan root, hanya tumpukan bersarang dengan perubahan templat yang akan diperbarui.

Selain itu, keberadaan tumpukan bersarang dapat memengaruhi operasi pada tumpukan root. Misalnya, jika satu tumpukan bersarang macet dalam `UPDATE_ROLLBACK_IN_PROGRESS` status, tumpukan root akan menunggu hingga tumpukan bersarang itu menyelesaikan rollback-nya sebelum melanjutkan. Sebelum melanjutkan dengan operasi pembaruan, pastikan bahwa Anda memiliki izin IAM untuk membatalkan pembaruan tumpukan jika itu kembali. Untuk informasi selengkapnya, lihat [Kontrol CloudFormation akses dengan AWS Identity and Access Management](#).

Gunakan prosedur berikut untuk menemukan tumpukan root dan tumpukan bersarang.

Untuk melihat tumpukan akar suatu tumpukan nest

1. Masuk ke AWS Management Console dan buka AWS CloudFormation konsol di <https://console.aws.amazon.com/cloudformation>.
2. Pada halaman Stacks, pilih nama tumpukan bersarang yang ingin Anda lihat tumpukan root.

Tumpukan bersarang menampilkan NESTED di atas nama tumpukannya.

3. Pada tab Stack info, di bagian Ikhtisar, pilih nama tumpukan yang terdaftar sebagai tumpukan Root.

Untuk melihat tumpukan nest yang dimiliki suatu tumpukan akar

1. Dari tumpukan root yang tumpukan bersarangnya ingin Anda lihat, pilih tab Resources.
2. Di kolom Type, cari sumber daya tipe `AWS::CloudFormation::Stack`.

Informasi terkait

- [Membuat nest tumpukan yang ada](#)
- [Memahami perilaku pembaruan sumber daya tumpukan](#)
- [Lanjutkan memutar kembali dari pembaruan tumpukan bersarang yang gagal](#)
- [Kegagalan rollback tumpukan bersarang](#)

Buat kondisi tunggu di CloudFormation template

Topik ini menjelaskan cara membuat kondisi tunggu dalam template untuk mengoordinasikan pembuatan sumber daya tumpukan atau melacak kemajuan proses konfigurasi. Misalnya, Anda dapat memulai pembuatan sumber daya lain setelah konfigurasi aplikasi selesai sebagian, atau Anda dapat mengirim sinyal selama proses instalasi dan konfigurasi untuk melacak kemajuannya.

Saat CloudFormation membuat tumpukan yang menyertakan kondisi tunggu:

- Ini menciptakan kondisi tunggu seperti sumber daya lainnya dan menetapkan status kondisi tunggu ke `CREATE_IN_PROGRESS`.
- CloudFormation menunggu sampai menerima jumlah sinyal keberhasilan yang diperlukan atau periode batas waktu tunggu telah kedaluwarsa.
- Jika menerima jumlah sinyal keberhasilan yang diperlukan sebelum periode batas waktu berakhir:
 - Status kondisi tunggu berubah menjadi `CREATE_COMPLETE`
 - Pembuatan tumpukan berlanjut
- Jika batas waktu berakhir atau sinyal kegagalan diterima:
 - Status kondisi tunggu berubah menjadi `CREATE_FAILED`
 - Tumpukan gulungan kembali

⚠ Important

Untuk sumber daya Amazon EC2 dan Auto Scaling, sebaiknya gunakan CreationPolicy atribut alih-alih kondisi tunggu. Tambahkan CreationPolicy atribut ke sumber daya tersebut, dan gunakan skrip pembantu sinyal cfn untuk memberi sinyal ketika proses pembuatan instance telah berhasil diselesaikan.

Untuk informasi lebih lanjut, lihat [Atribut CreationPolicy](#).

ℹ Note

Jika Anda menggunakan AWS PrivateLink, sumber daya di VPC yang merespons kondisi tunggu harus memiliki akses ke bucket Amazon Simple Storage Service (Amazon S3) CloudFormation khusus. Sumber daya harus mengirimkan respons kondisi tunggu ke URL Amazon S3 yang telah ditetapkan sebelumnya. Jika mereka tidak dapat mengirim tanggapan ke Amazon S3, tidak CloudFormation akan menerima respons dan operasi tumpukan gagal. Untuk informasi selengkapnya, lihat [Akses CloudFormation menggunakan endpoint antarmuka \(\)AWS PrivateLink](#) dan [Mengontrol akses dari titik akhir VPC dengan](#) kebijakan bucket.

Topik

- [Membuat kondisi tunggu di template Anda](#)
- [Tunggu sintaks sinyal kondisi](#)
- [Mengakses data sinyal](#)

Membuat kondisi tunggu di template Anda**1. Tunggu kondisi pegangan**

Anda mulai dengan mendefinisikan [AWS::CloudFormation::WaitConditionHandle](#) sumber daya dalam template stack. Sumber daya ini menghasilkan URL presigned yang diperlukan untuk mengirim sinyal. Ini memungkinkan Anda untuk mengirim sinyal tanpa harus memberikan AWS kredensi Anda. Misalnya:

Resources :

MyWaitHandle:

Type: `AWS::CloudFormation::WaitConditionHandle`

2. Kondisi menunggu

Selanjutnya, Anda menentukan [AWS::CloudFormation::WaitCondition](#) sumber daya dalam template stack. Struktur dasar dari `AWS::CloudFormation::WaitCondition` terlihat seperti ini:

MyWaitCondition:

Type: `AWS::CloudFormation::WaitCondition`

Properties:

Handle: *String*

Timeout: *String*

Count: *Integer*

Sumber `AWS::CloudFormation::WaitCondition` daya memiliki dua properti yang diperlukan dan satu properti opsional.

- **Handle(wajib)** - Referensi ke yang `WaitConditionHandle` dideklarasikan dalam template.
- **Timeout(wajib)** — Jumlah detik CloudFormation untuk menunggu jumlah sinyal yang diperlukan untuk diterima. Timeout adalah properti terikat minimum, yang berarti batas waktu terjadi tidak lebih cepat dari waktu yang Anda tentukan, tetapi dapat terjadi segera setelahnya. Waktu maksimum yang dapat Anda tentukan adalah 43.200 detik (12 jam).
- **Count(opsional)** — Jumlah sinyal sukses yang CloudFormation harus diterima sebelum menetapkan status kondisi tunggu itu `CREATE_COMPLETE` dan melanjutkan pembuatan tumpukan. Jika tidak ditentukan, nilai defaultnya adalah 1.

Biasanya, Anda ingin kondisi menunggu dimulai segera setelah pembuatan sumber daya tertentu. Anda melakukan ini dengan menambahkan `DependsOn` atribut ke kondisi tunggu. Saat Anda menambahkan `DependsOn` atribut ke kondisi tunggu, CloudFormation buat sumber daya di `DependsOn` atribut terlebih dahulu, lalu buat kondisi tunggu. Untuk informasi lebih lanjut, lihat [Atribut DependsOn](#).

Contoh berikut menunjukkan kondisi tunggu yang:

- Dimulai setelah penciptaan `MyEC2Instance` sumber daya yang sukses
- Menggunakan `MyWaitHandle` sumber daya sebagai `WaitConditionHandle`
- Memiliki batas waktu 4500 detik

- Memiliki default Count 1 (karena tidak ada Count properti yang ditentukan)

MyWaitCondition:

```
Type: AWS::CloudFormation::WaitCondition
DependsOn: MyEC2Instance
Properties:
  Handle: !Ref MyWaitHandle
  Timeout: '4500'
```

3. Mengirim sinyal

Untuk memberi sinyal keberhasilan atau kegagalan CloudFormation, Anda biasanya menjalankan beberapa kode atau skrip. Misalnya, aplikasi yang berjalan pada EC2 instance mungkin melakukan beberapa tugas konfigurasi tambahan dan kemudian mengirim sinyal CloudFormation untuk menunjukkan penyelesaian.

Sinyal harus dikirim ke URL yang telah ditentukan sebelumnya yang dihasilkan oleh pegangan kondisi tunggu. Anda menggunakan URL yang telah ditentukan sebelumnya untuk memberi sinyal keberhasilan atau kegagalan.

Untuk mengirim sinyal

1. Untuk mengambil URL presigned dalam template, gunakan fungsi Ref intrinsik dengan nama logis dari handle kondisi tunggu.

Seperti yang ditunjukkan pada contoh berikut, template Anda dapat mendeklarasikan EC2 instance Amazon dan meneruskan URL yang telah ditetapkan sebelumnya ke EC2 instance menggunakan properti Amazon. EC2 UserData Hal ini memungkinkan skrip atau aplikasi yang berjalan pada instance tersebut untuk menandakan keberhasilan atau kegagalan.

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" }
```

Ini menghasilkan UserData output yang mirip dengan:

```
SignalURL=https://amzn-s3-demo-bucket.s3.amazonaws.com/...
```

Catatan: Dalam AWS Management Console dan alat baris perintah, URL presigned ditampilkan sebagai ID fisik dari sumber daya penanganan kondisi tunggu.

2. (Opsional) Untuk mendeteksi kapan tumpukan memasuki kondisi tunggu, Anda dapat menggunakan salah satu metode berikut:
 - Jika Anda membuat tumpukan dengan pemberitahuan yang diaktifkan, CloudFormation mengeluarkan pemberitahuan untuk setiap peristiwa tumpukan untuk topik tertentu. Jika Anda atau aplikasi Anda berlangganan topik itu, Anda dapat memantau pemberitahuan untuk peristiwa pembuatan penanganan kondisi tunggu dan mengambil URL yang telah ditandatangani dari pesan pemberitahuan.
 - Anda juga dapat memantau peristiwa tumpukan menggunakan AWS Management Console, the AWS CLI, atau SDK.
3. Untuk mengirim sinyal, Anda mengirim pesan permintaan HTTP menggunakan URL yang telah ditandatangani. Metode permintaan harus PUT dan Content-Type header harus berupa string kosong atau dihilangkan. Pesan permintaan harus berupa struktur formulir JSON yang ditentukan dalam [Tunggu sintaks sinyal kondisi](#).

Anda harus mengirim jumlah sinyal sukses yang ditentukan oleh Count properti agar dapat CloudFormation melanjutkan pembuatan tumpukan. Jika Anda memiliki Count yang lebih besar dari 1, UniqueId nilai untuk setiap sinyal harus unik di semua sinyal yang dikirim ke kondisi tunggu tertentu. UniqueIdIni adalah string alfanumerik arbitrer.

curlPerintah adalah salah satu cara untuk mengirim sinyal. Contoh berikut menunjukkan baris curl perintah yang menandakan keberhasilan untuk kondisi menunggu.

```
$ 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"
```

di mana file `/tmp/a` berisi struktur JSON berikut:

```
{
  "Status" : "SUCCESS",
  "Reason" : "Configuration Complete",
  "UniqueId" : "ID1234",
  "Data" : "Application has completed configuration."
}
```

Contoh ini menunjukkan baris `curl` perintah yang mengirimkan sinyal sukses yang sama kecuali mengirimkan struktur JSON sebagai parameter pada baris perintah.

```
$ 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"
```

Tunggu sintaks sinyal kondisi

Saat Anda mengirim sinyal ke URL yang dihasilkan oleh pegangan kondisi tunggu, Anda harus menggunakan format JSON berikut:

```
{
  "Status" : "StatusValue",
  "UniqueId" : "Some UniqueId",
  "Data" : "Some Data",
  "Reason" : "Some Reason"
}
```

Properti

StatusBidang harus menjadi salah satu dari nilai berikut:

- SUCCESS
- FAILURE

UniqueIdBidang mengidentifikasi sinyal ke CloudFormation. Jika Count properti kondisi tunggu lebih besar dari 1, UniqueId nilainya harus unik di semua sinyal yang dikirim untuk kondisi tunggu tertentu; jika tidak, CloudFormation akan mempertimbangkan sinyal transmisi ulang dari sinyal yang dikirim sebelumnya dengan yang sama UniqueId dan mengabaikannya.

DataBidang dapat berisi informasi apa pun yang ingin Anda kirim kembali dengan sinyal. Anda dapat mengakses Data nilai dengan menggunakan GetAtt fungsi [Fn::](#) dalam template.

ReasonBidang ini adalah string tanpa batasan lain pada kontennya selain kepatuhan JSON.

Mengakses data sinyal

Untuk mengakses data yang dikirim oleh sinyal yang valid, Anda dapat membuat nilai output untuk kondisi tunggu di CloudFormation template Anda. Misalnya:

Outputs:

WaitConditionData:

Description: *The data passed back as part of signalling the WaitCondition*

Value: !GetAtt *MyWaitCondition*.Data

Anda kemudian dapat melihat data ini menggunakan [describe-stacks](#) perintah, atau tab Output CloudFormation konsol.

Fn::GetAttFungsi mengembalikan UniqueId dan Data sebagai name/value pasangan dalam struktur JSON. Misalnya:

```
{"Signal1": "Application has completed configuration."}
```

Buat konfigurasi sumber daya yang dapat digunakan kembali yang dapat disertakan di seluruh templat dengan modul CloudFormation

Modul adalah cara Anda untuk mengemas konfigurasi sumber daya untuk dimasukkan di seluruh tumpukan templat, dengan cara yang transparan, mudah dikelola, dan berulang. Modul dapat merangkum konfigurasi layanan umum dan praktik terbaik sebagai blok bangunan modular yang

dapat disesuaikan untuk Anda sertakan dalam templat tumpukan Anda. Modul memungkinkan Anda untuk menyertakan konfigurasi sumber daya yang menggabungkan praktik terbaik, pengetahuan domain ahli, dan pedoman yang diterima (untuk area seperti keamanan, kepatuhan, tata kelola, dan peraturan industri) dalam templat Anda, tanpa harus memiliki pengetahuan mendalam tentang seluk-beluk sumber daya implementasi.

Misalnya, ahli domain dalam jaringan dapat membuat modul yang berisi grup keamanan internal dan aturan masukan/keluar yang mematuhi pedoman keamanan. Anda kemudian dapat memasukkan modul itu ke dalam template Anda untuk menyediakan infrastruktur jaringan yang aman di tumpukan Anda—tanpa harus menghabiskan waktu mencari tahu bagaimana VPCs, subnet, grup keamanan, dan gateway bekerja. Dan karena modul berversi, jika pedoman keamanan berubah dari waktu ke waktu, penulis modul dapat membuat versi baru dari modul yang menggabungkan perubahan-perubahan tersebut.

Karakteristik penggunaan modul dalam templat meliputi:

- **Prediktabilitas** — Modul harus mematuhi skema yang didaftarkan di CloudFormation registri, sehingga Anda tahu sumber daya apa yang dapat diselesaikan setelah Anda memasukkannya ke dalam templat Anda.
- **Reusability** - Anda dapat menggunakan modul yang sama di beberapa template dan akun.
- **Keterlacakan** — CloudFormation mempertahankan pengetahuan tentang sumber daya mana dalam tumpukan yang disediakan dari modul, memungkinkan Anda untuk dengan mudah memahami sumber perubahan sumber daya.
- **Pengelolaan** - Setelah Anda mendaftarkan modul, Anda dapat mengelolanya melalui CloudFormation registri, termasuk pembuatan versi dan akun dan ketersediaan regional.

Sebuah modul dapat berisi:

- Satu sumber daya atau lebih yang akan ditetapkan dari modul, bersama dengan data yang terkait, seperti output atau kondisi.
- Semua parameter modul, yang memungkinkan Anda untuk menentukan nilai-nilai kustom setiap kali modul digunakan.

Untuk informasi tentang pengembangan modul, lihat [Mengembangkan modul](#) di Panduan Pengguna CloudFormation CLI.

Topik

- [Pertimbangan saat menggunakan modul](#)
- [Memahami pembuatan versi modul](#)
- [Gunakan modul dari registri CloudFormation pribadi](#)
- [Gunakan parameter untuk menentukan nilai modul](#)
- [Sumber daya modul referensi dalam CloudFormation templat](#)

Pertimbangan saat menggunakan modul

- Tidak ada biaya tambahan untuk menggunakan modul. Anda hanya membayar untuk sumber daya yang diselesaikan modul tersebut di tumpukan Anda.
- CloudFormation kuota, seperti jumlah maksimum sumber daya yang diizinkan dalam tumpukan, atau ukuran maksimum badan templat, berlaku untuk templat yang diproses apakah sumber daya yang disertakan dalam templat itu berasal dari modul atau tidak. Untuk informasi selengkapnya, lihat [Memahami CloudFormation kuota](#).
- Tag yang Anda tentukan di tingkat tumpukan ditugaskan ke sumber daya individu yang berasal dari modul.
- Skrip pembantu yang ditentukan pada tingkat modul tidak menyebar ke sumber daya individu yang terkandung dalam modul saat CloudFormation memproses template.
- Output yang ditentukan dalam modul disebarkan ke output pada tingkat templat.

Setiap output akan diberikan ID logis yang merupakan rangkaian dari nama logis modul dan nama output seperti yang didefinisikan dalam modul. Untuk informasi selengkapnya, lihat [Dapatkan output yang diekspor dari tumpukan yang diterapkan CloudFormation](#).

- Parameter yang ditentukan dalam modul tidak disebarkan ke parameter pada tingkat templat.

Namun, Anda dapat membuat parameter tingkat templat yang mereferensikan parameter tingkat modul. Untuk informasi selengkapnya, lihat [Gunakan parameter untuk menentukan nilai modul](#).

Memahami pembuatan versi modul

CloudFormation Registri bertindak sebagai repositori tempat Anda dapat mendaftar dan mengelola modul untuk digunakan di dalam Akun AWS dan Wilayah Anda. Anda dapat mendaftarkan modul dari berbagai sumber, termasuk AWS, penerbit pihak ketiga, dan ekstensi kustom Anda sendiri, dalam akun dan Wilayah Anda. Untuk informasi selengkapnya, lihat [Mengelola ekstensi dengan CloudFormation registri](#).

Modul dapat memiliki versi yang berbeda, sehingga Anda dapat menentukan versi modul yang ingin Anda gunakan. Kemampuan pembuatan versi ini sangat berguna ketika Anda perlu memperbarui atau memodifikasi modul tanpa merusak tumpukan yang ada yang bergantung padanya.

Ingatlah pertimbangan berikut saat menggunakan beberapa versi modul:

- Selama operasi tumpukan, CloudFormation gunakan versi modul apa pun yang saat ini terdaftar sebagai versi default di Akun AWS dan Wilayah tempat operasi tumpukan sedang dilakukan. Ini termasuk modul yang bersarang di modul lain.

Oleh karena itu, ketahuilah bahwa jika Anda memiliki versi berbeda dari modul yang sama yang terdaftar sebagai versi default di akun atau Wilayah yang berbeda, menggunakan templat yang sama dapat menghasilkan hasil yang berbeda.

- Selama operasi tumpukan, CloudFormation gunakan versi sumber daya apa pun yang saat ini terdaftar sebagai versi default di Akun AWS dan Wilayah tempat operasi tumpukan sedang dilakukan. Ini termasuk sumber daya yang dihasilkan dengan menyertakan modul.
- Mengubah versi default dari modul tidak memulai setiap operasi pembaruan tumpukan. Namun, lain kali Anda melakukan operasi tumpukan dengan template apa pun yang berisi modul itu, seperti pembaruan tumpukan, CloudFormation akan menggunakan versi default baru dalam operasi.

Satu pengecualian untuk ini adalah melakukan pembaruan tumpukan dengan menggunakan templat sebelumnya yang sudah ditentukan, seperti yang dijelaskan di bawah ini.

- Untuk operasi pembaruan tumpukan, jika Anda menentukan opsi gunakan templat sebelumnya, CloudFormation gunakan templat yang diproses sebelumnya untuk pembaruan tumpukan, dan tidak memproses ulang modul untuk perubahan apa pun yang mungkin Anda buat.
- Untuk menjamin hasil yang seragam, jika Anda menyertakan modul dalam template tumpukan untuk digunakan dengan kumpulan tumpukan, Anda harus memastikan bahwa versi modul yang sama ditetapkan sebagai versi default di semua akun dan Wilayah tempat Anda berencana untuk menerapkan instance tumpukan Anda. Ini termasuk untuk modul yang bersarang di modul lain. Untuk informasi selengkapnya, lihat [Mengelola tumpukan di seluruh akun dan Wilayah dengan StackSets](#).

Persyaratan untuk mengaktifkan modul publik pihak ketiga

Agar berhasil mengaktifkan modul publik pihak ketiga di akun dan Wilayah Anda, berikut ini harus benar untuk setiap ekstensi publik pihak ketiga (sumber daya atau modul) yang disertakan dalam modul:

- Aktivasi ekstensi — Ekstensi harus diaktifkan di akun dan Wilayah tempat Anda ingin menggunakannya. Untuk informasi selengkapnya, lihat [Gunakan ekstensi publik pihak ketiga dari CloudFormation registri](#).
- Pendaftaran alias - Jika ekstensi dalam modul menggunakan alias nama tipe, ekstensi harus terdaftar di akun Anda dan Wilayah menggunakan nama jenis alias yang sama. Untuk informasi selengkapnya, lihat [Gunakan alias untuk merujuk ke ekstensi](#).
- Kompatibilitas versi — Versi ekstensi yang saat ini diaktifkan harus menjadi salah satu versi utama yang didukung dari ekstensi yang ditentukan dalam modul.

Jika Anda tidak memiliki ekstensi publik pihak ketiga yang benar dan versi ekstensi diaktifkan, CloudFormation akan gagal operasi dengan kesalahan daftar ekstensi dan versi yang perlu diaktifkan sebelum modul dapat berhasil diaktifkan.

Gunakan modul dari registri CloudFormation pribadi

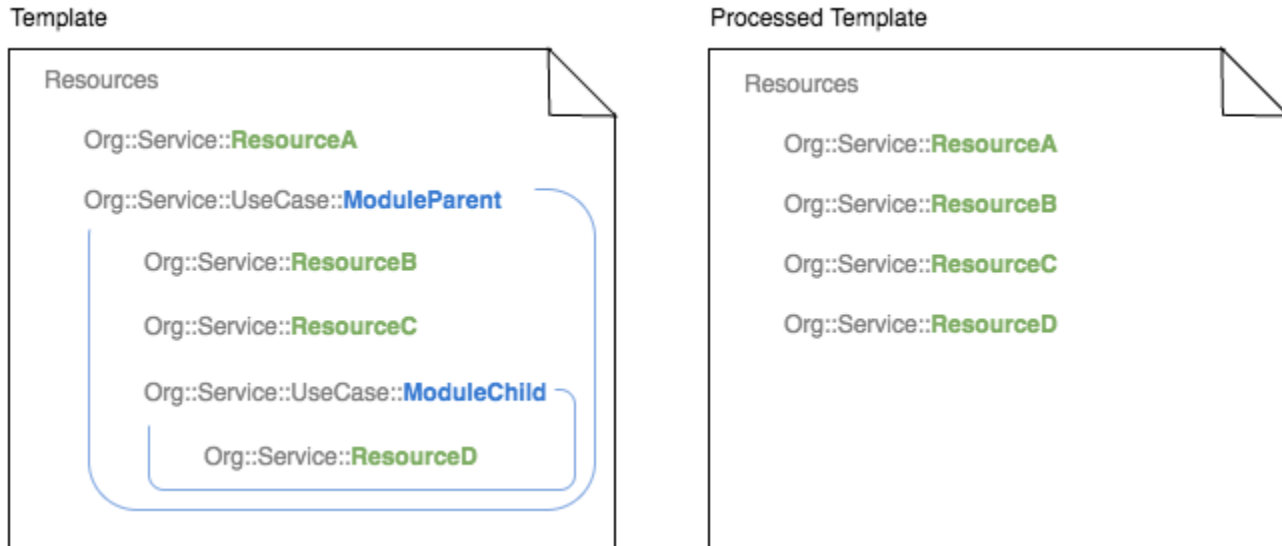
Topik ini menjelaskan cara menggunakan modul dalam CloudFormation template. Pikirkan modul sebagai bundel sumber daya yang telah dibuat sebelumnya yang dapat Anda tambahkan ke templat Anda.

Untuk menggunakan modul, langkah-langkahnya adalah sebagai berikut:

- Daftarkan modul — Anda mendaftarkan modul dalam CloudFormation registri sebagai ekstensi pribadi. Pastikan itu terdaftar di Akun AWS dan Wilayah tempat Anda bekerja. Untuk informasi selengkapnya, lihat [CloudFormation konsep registri](#).
- Sertakan dalam template Anda - Tambahkan modul ke [Resources](#) bagian CloudFormation template Anda, seperti yang Anda lakukan dengan sumber daya lainnya. Anda juga harus menyediakan properti yang diperlukan untuk modul.
- Buat atau perbarui tumpukan — Saat Anda memulai operasi tumpukan, buat templat yang CloudFormation diproses yang menyelesaikan modul apa pun yang disertakan ke dalam sumber daya yang sesuai.
- Pratinjau perubahan — Sebelum membuat perubahan, Anda dapat menggunakan set perubahan untuk melihat sumber daya apa yang akan ditambahkan atau diubah. Untuk informasi selengkapnya, lihat [Perbarui CloudFormation tumpukan menggunakan set perubahan](#).

Perhatikan contoh berikut: Anda memiliki templat yang berisi sumber daya dan modul. Template berisi satu sumber daya `individuResourceA`, serta modul `ModuleParent`. Modul itu berisi

dua sumber daya `ResourceC`, `ResourceB` dan, serta modul bersarang, `ModuleChild`. `ModuleChild` berisi satu sumber daya, `ResourceD`. Jika Anda membuat tumpukan dari template ini, CloudFormation proses template dan menyelesaikan modul ke sumber daya yang sesuai. Tumpukan yang dihasilkan memiliki empat sumber daya: `ResourceA`, `ResourceB`, `ResourceC`, dan `ResourceD`.



CloudFormation melacak sumber daya mana dalam tumpukan yang dibuat dari modul. Anda dapat melihat informasi ini di Peristiwa, Sumber Daya, dan tab Drifts untuk tumpukan tertentu, dan itu juga termasuk dalam perubahan set pratinjau.

Modul dibedakan dari sumber daya dalam template karena mereka mematuhi empat bagian konvensi penamaan berikut, sebagai kebalikan dari konvensi tiga bagian khas yang digunakan oleh sumber daya:

```
organization::service::use-case::MODULE
```

Gunakan parameter untuk menentukan nilai modul

Di CloudFormation, Anda dapat menggunakan parameter template untuk menyesuaikan tumpukan Anda dengan memberikan nilai input selama pembuatan atau pembaruan tumpukan. Parameter ini memungkinkan Anda untuk mengubah aspek-aspek tertentu dari tumpukan berdasarkan kebutuhan Anda. Untuk informasi selengkapnya tentang mendefinisikan parameter template, lihat [CloudFormation Parameters sintaks template](#).

Demikian pula, modul juga dapat memiliki parameter. Parameter modul ini memungkinkan Anda untuk memasukkan nilai kustom ke modul dari template (atau modul lain) yang menggunakannya.

Modul kemudian dapat menggunakan nilai-nilai kustom ini untuk menetapkan nilai properti untuk sumber daya yang dikandungnya.

Anda juga dapat menentukan parameter template yang mengatur properti modul, sehingga Anda dapat memasukkan nilai yang diteruskan ke modul pada saat operasi tumpukan.

Jika modul berisi modul bersarang yang memiliki parameter modulnya sendiri, Anda dapat:

- Menentukan nilai untuk parameter modul bersarang ini langsung di modul induk.
- Mendefinisikan parameter modul yang sesuai dalam modul induk yang memungkinkan parameter modul bersarang untuk ditetapkan oleh templat (atau modul) di mana modul induk berada.

Menggunakan parameter templat untuk menentukan nilai parameter modul

Contoh berikut menunjukkan cara menentukan parameter templat yang lulus nilai modul.

Template ini berisi `My::S3::SampleBucket::MODULE` mendefinisikan parameter template, `BucketName`, yang memungkinkan pengguna untuk menentukan nama bucket S3 selama operasi stack.

```
# 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
```

Menentukan properti pada sumber daya dalam modul anak dari modul induk

Contoh berikut menggambarkan cara menentukan nilai parameter dalam modul yang bersarang di modul lain.

Modul pertama ini, `My::S3::SampleBucketPrivate::MODULE`, akan menjadi modul anak. Ini mendefinisikan dua parameter: `BucketName` dan `AccessControl`. Nilai-nilai yang ditentukan untuk parameter ini digunakan untuk menentukan `BucketName` dan `AccessControl` properti

`AWS::S3::Bucket` sumber daya yang dimuat modul. Berikut adalah fragmen templat untuk `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
```

Selanjutnya, modul sebelumnya bersarang dalam modul induk, `My::S3::SampleBucket::MODULE`. Modul induk, `My::S3::SampleBucket::MODULE`, menetapkan parameter modul anak dengan cara berikut:

- Ini menetapkan `AccessControl` parameter `My::S3::SampleBucketPrivate::MODULE` untuk `Private`.
- Untuk `BucketName`, ia mendefinisikan parameter modul, yang akan memungkinkan nama bucket untuk ditentukan dalam templat (atau modul) yang berisi `My::S3::SampleBucket::MODULE`.

```
# 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
```

Menentukan kendala untuk parameter modul

Parameter modul tidak mendukung penegakan kendala. Untuk melakukan pemeriksaan kendala pada parameter modul, buat parameter template dengan batasan yang diinginkan. Kemudian, referensikan parameter template itu dalam parameter modul Anda. Untuk informasi selengkapnya tentang mendefinisikan parameter template, lihat [CloudFormation Parameterssintaks template](#).

Sumber daya modul referensi dalam CloudFormation templat

Dalam CloudFormation template, Anda sering perlu mengatur properti pada satu sumber daya berdasarkan nama atau properti sumber daya lain. Untuk informasi selengkapnya, lihat [Referensi sumber daya](#).

Untuk mereferensikan sumber daya yang terkandung dalam modul dalam CloudFormation template Anda, Anda harus menggabungkan dua nama logis:

- Nama logis yang Anda berikan ke modul itu sendiri ketika Anda memasukkannya ke dalam template Anda.
- Nama logis dari sumber daya tertentu dalam modul itu.

Anda dapat menggabungkan dua nama logis ini dengan atau tanpa menggunakan titik (.) di antara keduanya. Misalnya, jika nama logis modul adalah `MyModule` dan nama logis sumber daya adalah `MyBucket`, Anda dapat merujuk ke sumber daya itu sebagai salah satu `MyModule.MyBucket` atau `MyModuleMyBucket`.

Untuk menemukan nama logis sumber daya di dalam modul, Anda dapat berkonsultasi dengan skema modul, yang tersedia di CloudFormation registri atau dengan menggunakan [DescribeType](#) operasi. Skema ini mencantumkan semua sumber daya dan nama logisnya yang merupakan bagian dari modul.

Setelah Anda memiliki nama logis lengkap, Anda dapat menggunakan CloudFormation fungsi seperti `GetAtt` dan `Ref` untuk mengakses nilai properti pada sumber daya modul.

Misalnya, Anda memiliki `My::S3::SampleBucket::MODULE` modul yang berisi `AWS::S3::Bucket` sumber daya dengan nama logis `S3Bucket`. Untuk merujuk ke nama bucket ini

menggunakan Ref fungsi, Anda menggabungkan nama modul dalam template (MyBucket) Anda dengan nama logis sumber daya dalam modul (S3Bucket). Nama logis lengkapnya adalah salah satu MyBucket.S3Bucket atau MyBucketS3Bucket.

Contoh Templat

Contoh template berikut membuat bucket S3 menggunakan `My::S3::SampleBucket::MODULE` modul. Itu juga membuat antrian Amazon SQS dan mengatur namanya menjadi sama dengan nama bucket dari modul. Selain itu, template menampilkan Amazon Resource Name (ARN) dari bucket S3 yang dibuat.

```
# 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
```


Mengelola AWS sumber daya sebagai satu unit dengan AWS CloudFormation tumpukan

Tumpukan adalah kumpulan sumber AWS daya yang dapat Anda kelola sebagai satu unit. Dengan kata lain, Anda dapat membuat, memperbarui, dan menghapus kumpulan sumber daya dengan membuat, memperbarui, dan menghapus tumpukan.

Membuat tumpukan melibatkan penerapan CloudFormation template yang menentukan sumber daya dan konfigurasinya, yang CloudFormation kemudian menyediakan dan mengonfigurasi.

Memperbarui tumpukan melibatkan membuat perubahan pada template atau parameter.

CloudFormation membandingkan perubahan yang Anda kirimkan dengan status tumpukan Anda saat ini dan hanya memperbarui sumber daya yang diubah. CloudFormation mungkin mengganggu sumber daya atau mengganti sumber daya yang diperbarui, tergantung pada properti yang Anda perbarui. Untuk informasi selengkapnya tentang perilaku pembaruan sumber daya, lihat [Memahami perilaku pembaruan sumber daya tumpukan](#).

CloudFormation menyediakan dua metode untuk memperbarui tumpukan:

- **Ubah set** - Dengan set perubahan, Anda dapat melihat pratinjau perubahan yang CloudFormation akan dilakukan pada tumpukan Anda, dan kemudian memutuskan apakah akan menerapkan perubahan tersebut. Set perubahan adalah dokumen berformat JSON yang merangkum perubahan yang CloudFormation akan dilakukan ke tumpukan. Gunakan set perubahan saat Anda ingin memastikan bahwa CloudFormation tidak membuat perubahan yang tidak disengaja atau ketika Anda ingin mempertimbangkan beberapa opsi. Misalnya, Anda dapat menggunakan set perubahan untuk memverifikasi bahwa CloudFormation tidak akan menggantikan instance database stack Anda selama pembaruan.
- **Pembaruan langsung** - Saat Anda memperbarui tumpukan secara langsung, Anda mengirimkan perubahan dan CloudFormation segera menerapkannya. Gunakan pembaruan langsung jika Anda ingin men-deploy pembaruan dengan cepat.

Menghapus tumpukan menghapus sumber daya yang terkait dengannya. Tumpukan, untuk instans, dapat mencakup semua sumber daya yang diperlukan untuk menjalankan aplikasi web, seperti server web, basis data, dan aturan jaringan. Jika Anda tidak lagi memerlukan aplikasi web tersebut, Anda cukup menghapus tumpukan, dan semua sumber daya terkaitnya akan dihapus.

 Note

Anda akan dikenakan biaya untuk sumber daya tumpukan berdasarkan waktu beroperasi (meski Anda langsung menghapus tumpukan).

CloudFormation memastikan semua sumber daya tumpukan dibuat atau dihapus sebagaimana mestinya. Karena CloudFormation memperlakukan sumber daya tumpukan sebagai unit tunggal, semuanya harus berhasil dibuat atau dihapus agar tumpukan dapat dibuat atau dihapus. Jika sumber daya tidak dapat dibuat, CloudFormation membatalkan tumpukan dan secara otomatis menghapus sumber daya yang dibuat. Jika sumber daya tidak dapat dihapus, sumber daya yang tersisa dipertahankan hingga tumpukan berhasil dihapus.

Topik

- [Antarmuka untuk mengelola tumpukan Anda](#)
- [Buat tumpukan dari CloudFormation konsol](#)
- [Lihat informasi tumpukan dari CloudFormation konsol](#)
- [Perbarui template tumpukan Anda](#)
- [Memahami perilaku pembaruan sumber daya tumpukan](#)
- [Perbarui CloudFormation tumpukan menggunakan set perubahan](#)
- [Perbarui tumpukan secara langsung](#)
- [Batalkan pembaruan tumpukan](#)
- [Hapus tumpukan dari CloudFormation konsol](#)
- [Pantau kemajuan tumpukan](#)
- [Gulung kembali CloudFormation tumpukan Anda pada pelanggaran alarm dengan pemicu rollback](#)
- [Mendeteksi perubahan konfigurasi yang tidak terkelola pada tumpukan dan sumber daya dengan deteksi drift](#)
- [Impor AWS sumber daya ke CloudFormation tumpukan](#)
- [Stack refactoring](#)
- [Dukungan jenis sumber daya](#)
- [Gunakan tautan buat cepat untuk membuat tumpukan CloudFormation](#)
- [Contoh perintah operasi CloudFormation tumpukan untuk AWS CLI dan PowerShell](#)

Antarmuka untuk mengelola tumpukan Anda

Anda dapat mengelola CloudFormation tumpukan menggunakan antarmuka berikut:

- CloudFormation konsol — Menyediakan antarmuka web yang dapat Anda gunakan untuk mengakses tumpukan Anda. Anda dapat mengakses CloudFormation konsol dengan masuk ke AWS Management Console, menggunakan kotak pencarian di bilah navigasi untuk mencari CloudFormation, dan kemudian memilih CloudFormation dari hasil pencarian.
- AWS Command Line Interface Menyediakan perintah untuk serangkaian AWS layanan yang luas, termasuk CloudFormation, dan didukung pada Windows, Mac, dan Linux. Untuk informasi tentang CloudFormation perintah, lihat [cloudformation](#) di Referensi AWS CLI Perintah.
- Alat AWS untuk PowerShell— Satu set PowerShell modul yang dibangun di atas fungsionalitas yang diekspose oleh SDK untuk .NET. Alat untuk PowerShell memungkinkan Anda melakukan operasi skrip pada AWS sumber daya Anda dari PowerShell baris perintah. Anda dapat menemukan cmdlet untuk CloudFormation di Referensi [Alat AWS untuk PowerShell Cmdlet](#).
- Query API - Menyediakan tindakan API tingkat rendah yang Anda panggil menggunakan permintaan HTTPS. Jika Anda melakukan panggilan API dalam aplikasi Anda, Anda harus menulis kode untuk menangani detail tingkat rendah, seperti membuat hash untuk menandatangani permintaan. Untuk informasi selengkapnya tentang tindakan API CloudFormation, lihat [Tindakan](#) di Referensi AWS CloudFormation API.
- AWS SDKs Menyediakan bahasa khusus APIs dan menangani banyak detail koneksi, seperti menghitung tanda tangan, menangani percobaan ulang permintaan, dan penanganan kesalahan. Untuk informasi selengkapnya, lihat [Alat untuk Membangun AWS](#).
- AWS Cloud Development Kit (AWS CDK)— AWS CDK Ini adalah kerangka pengembangan perangkat lunak open-source yang memungkinkan Anda untuk mendefinisikan AWS infrastruktur menggunakan bahasa pemrograman yang sudah dikenal seperti TypeScript, Python, Java, dan .NET. Dengan CDK, Anda dapat memodelkan sumber daya aplikasi Anda dan kemudian menyediakannya menggunakan CloudFormation langsung dari lingkungan pengembangan terintegrasi (IDE) Anda. Lihat informasi yang lebih lengkap di [AWS Cloud Development Kit \(AWS CDK\)](#).

Buat tumpukan dari CloudFormation konsol

Anda dapat membuat template tumpukan dan kemudian menggunakannya untuk membuat tumpukan menggunakan CloudFormation konsol atau alat baris perintah. Konsol menyediakan antarmuka yang

digerakkan oleh penyihir dengan opsi yang telah ditentukan, yang merampingkan proses pembuatan tumpukan.

Topik

- [Membuat tumpukan](#)
- [Konfigurasi opsi tumpukan](#)
- [Pratinjau konfigurasi tumpukan Anda](#)

Membuat tumpukan

Ikuti langkah-langkah di bagian ini untuk menyebarkan template Anda dan membuat tumpukan.

Prasyarat

- Anda harus telah membuat template tumpukan. Untuk informasi selengkapnya, lihat [Bekerja dengan CloudFormation template](#).

Untuk membuat tumpukan (konsol)

1. Buka AWS CloudFormation konsol di <https://console.aws.amazon.com/cloudformation>.
2. Pada bilah navigasi di bagian atas layar, pilih Wilayah AWS untuk membuat tumpukan.
3. Pada halaman Stacks, pilih Buat tumpukan di kanan atas, lalu pilih Dengan sumber daya baru (standar).

Atau, Anda dapat memilih opsi Dengan sumber daya yang ada (sumber daya impor) untuk mengimpor AWS sumber daya yang ada yang dijelaskan dalam templat Anda. Untuk informasi lebih lanjut tentang opsi ini, lihat [Impor AWS sumber daya ke CloudFormation tumpukan](#).

4. Pada halaman Create stack, lakukan salah satu hal berikut:
 - Untuk menggunakan templat yang ada, untuk Prasyarat - Siapkan templat, pilih Pilih templat yang ada. Kemudian, di bawah Tentukan templat, pilih URL Amazon S3 atau Unggah file templat berdasarkan lokasi templat.
 - Jika Anda memilih URL Amazon S3, berikan URL ke file template dalam bucket S3.

Jika template Anda menyertakan tumpukan bersarang (misalnya, tumpukan yang dijelaskan dalam dokumen templat lain yang terletak di subdirektori), pastikan bucket S3 Anda berisi file dan direktori yang diperlukan.

Jika Anda memiliki template dari bucket berkemampuan versi, Anda dapat menentukan versi template tertentu dengan menambahkan `?versionId=version-id` ke URL. Untuk informasi selengkapnya tentang bucket berkemampuan versi, lihat [Bekerja dengan objek di bucket berkemampuan versi di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon](#).

URL harus mengarah ke template dengan ukuran maksimum 1 MB yang disimpan dalam bucket S3 yang telah Anda baca izin. URL bisa maksimal 1024 karakter. Beberapa sumber daya mungkin mengharuskan bucket berada di Wilayah yang sama dengan tumpukan.

- Jika Anda memilih Unggah file templat, pilih Pilih File untuk memilih file templat dari komputer lokal Anda. Ukuran file template harus 1 MB atau kurang.

Setelah Anda memilih template Anda, CloudFormation upload file dan menampilkan URL S3. CloudFormation mengunggahnya ke bucket Amazon S3 di bucket Anda. Akun AWS Jika Anda sudah memiliki bucket S3 yang dibuat oleh CloudFormation Anda Akun AWS, CloudFormation tambahkan template ke bucket itu. Jika Anda belum memiliki bucket CloudFormation -created yang sudah ada, bucket tersebut akan membuat bucket unik untuk setiap Region tempat Anda mengunggah file template.

Berikut ini adalah pertimbangan saat menggunakan bucket S3 yang dibuat oleh CloudFormation

- Bucket dapat diakses oleh siapa saja yang memiliki izin Amazon S3 di Anda. Akun AWS
- CloudFormation membuat bucket dengan enkripsi sisi server diaktifkan secara default, sehingga mengenkripsi semua objek yang disimpan dalam bucket.

Anda dapat langsung mengelola opsi enkripsi untuk bucket yang CloudFormation telah dibuat, misalnya, menggunakan konsol Amazon S3 <https://console.aws.amazon.com/s3/>di, atau. AWS CLI Untuk informasi selengkapnya, lihat [Menyetel perilaku enkripsi sisi server default untuk bucket Amazon S3 di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon](#).

- Anda dapat menggunakan bucket Anda sendiri dan mengelola izinnya dengan mengunggah templat secara manual ke Amazon S3. Saat Anda membuat atau memperbarui tumpukan, tentukan URL Amazon S3 dari file templat.
- Jika Anda tidak memiliki template siap, Anda dapat memilih Build from Infrastructure Composer untuk membuat template dengan Infrastructure Composer. Untuk informasi selengkapnya, lihat [Komposer Infrastruktur](#).

5. Pilih Berikutnya untuk melanjutkan dan memvalidasi template.

Sebelum melanjutkan, CloudFormation validasi template Anda untuk menangkap sintaksis dan beberapa kesalahan semantik, seperti dependensi melingkar. Selama validasi, periksa CloudFormation terlebih dahulu apakah template adalah JSON yang valid. Jika tidak, CloudFormation periksa apakah template tersebut adalah YAMB yang valid. Jika kedua pemeriksaan gagal, CloudFormation mengembalikan kesalahan validasi template.

6. Pada halaman Tentukan detail tumpukan, ketik nama tumpukan di kotak Nama tumpukan.

Nama tumpukan adalah pengidentifikasi yang membantu Anda menemukan tumpukan tertentu dari daftar tumpukan. Nama tumpukan hanya dapat berisi karakter alfanumerik (peka huruf besar/kecil) dan tanda hubung. Ini harus dimulai dengan karakter abjad dan tidak boleh lebih dari 128 karakter.

7. Di bagian Parameter, tentukan nilai untuk parameter yang ditentukan dalam templat.

8. Pilih Berikutnya untuk terus membuat tumpukan.

9. (Opsional) Pada halaman Configure stack options, ubah opsi stack default. Untuk informasi selengkapnya, lihat [Konfigurasi opsi tumpukan](#).

10. Jika template Anda berisi sumber daya IAM, untuk Kemampuan, pilih Saya mengakui bahwa templat ini dapat membuat sumber daya IAM untuk menentukan bahwa Anda ingin menggunakan sumber daya IAM dalam templat. Untuk informasi selengkapnya, lihat [Mengakui sumber daya IAM dalam templat CloudFormation](#).

11. Pilih Next untuk melanjutkan.

12. Pada halaman Tinjau dan buat, tinjau detail tumpukan Anda.

Untuk mengubah nilai apa pun sebelum meluncurkan tumpukan, pilih Edit pada bagian yang memiliki pengaturan yang ingin Anda ubah.

13. (Opsional) Anda dapat membuat set perubahan untuk melihat pratinjau konfigurasi tumpukan sebelum membuatnya. Pada halaman Tinjau dan buat, pilih Buat set perubahan dan ikuti petunjuknya. Untuk informasi selengkapnya, lihat [Pratinjau konfigurasi tumpukan Anda](#).

14. Pilih Kirim untuk meluncurkan tumpukan Anda.

CloudFormation kemudian akan melanjutkan untuk membuat semua sumber daya yang ditentukan dalam template.

Anda dapat memantau kemajuan dan status pembuatan tumpukan pada tab Acara untuk tumpukan baru Anda. Untuk informasi selengkapnya, lihat [Pantau kemajuan tumpukan](#).

Untuk membuat tumpukan menggunakan baris perintah

Anda dapat menggunakan salah satu perintah berikut:

- [create-stack](#) (AWS CLI)
- [New-CFNStack](#) (AWS Tools for Windows PowerShell)

Untuk contoh menggunakan baris perintah untuk membuat tumpukan, lihat [Contoh perintah operasi CloudFormation tumpukan untuk AWS CLI dan PowerShell](#).

Konfigurasi opsi tumpukan

Pada halaman Configure stack options, Anda dapat mengonfigurasi opsi untuk CloudFormation tumpukan Anda seperti tag, pemberitahuan peristiwa tumpukan, atau kebijakan tumpukan.

Anda dapat mengatur opsi tumpukan berikut:

Tanda

Anda dapat menambahkan hingga 50 pasangan kunci tag ke tumpukan Anda dan ke sumber daya apa pun yang CloudFormation mendukung penandaan. Tag adalah kunci dan nilai yang ditentukan pelanggan yang dapat ditetapkan ke AWS sumber daya untuk tujuan seperti pelacakan biaya.

Kunci terdiri dari karakter alfanumerik atau spasi. Kunci tanda dapat berisi hingga 127 karakter.

Nilai terdiri dari karakter alfanumerik atau spasi. Nilai tanda dapat berisi hingga 255 karakter.

Setelah pembuatan tumpukan, menambahkan, memperbarui, atau menghapus tag tingkat tumpukan akan memulai pembaruan tumpukan. Semua sumber daya yang mendukung propagasi tag tingkat tumpukan akan diperbarui sesuai dengan itu.

Izin

Peran layanan IAM yang ada yang CloudFormation dapat mengasumsikan. Alih-alih menggunakan kredensial akun Anda, CloudFormation gunakan kredensial peran untuk membuat tumpukan Anda. Untuk informasi selengkapnya, lihat [AWS CloudFormation peran layanan](#).

Opsi kegagalan tumpukan

Menentukan opsi kegagalan penyediaan untuk semua penerapan tumpukan dan mengubah operasi set. Untuk informasi selengkapnya, lihat [Pilih cara menangani kegagalan saat menyediakan sumber daya](#).

Opsi Roll back all stack resources akan mengembalikan semua sumber daya yang ditentukan dalam template saat status stack adalah CREATE_FAILED atau UPDATE_FAILED.

Untuk membuat operasi, opsi Pertahankan sumber daya yang berhasil disediakan mempertahankan status sumber daya yang berhasil, sementara sumber daya yang gagal akan tetap dalam keadaan gagal hingga operasi pembaruan berikutnya dilakukan.

Untuk operasi set pembaruan dan perubahan, opsi Pertahankan sumber daya yang berhasil disediakan mempertahankan status sumber daya yang berhasil sambil mengembalikan sumber daya yang gagal ke status stabil terakhir yang diketahui. Sumber daya yang gagal akan berada dalam UPDATE_FAILED keadaan. Sumber daya tanpa status stabil terakhir yang diketahui akan dihapus pada operasi tumpukan berikutnya.

Anda juga dapat mengatur opsi lanjutan berikut untuk pembuatan tumpukan:

Kebijakan tumpukan

Menetapkan sumber daya yang Anda ingin lindungi dari pembaruan yang tidak disengaja selama pembaruan tumpukan. Secara default, semua sumber daya dapat diperbarui selama pembaruan tumpukan.

Anda dapat memasukkan kebijakan tumpukan secara langsung sebagai JSON, atau mengunggah file JSON yang berisi kebijakan tumpukan. Untuk informasi selengkapnya, lihat [Mencegah pembaruan ke sumber daya tumpukan](#).

Konfigurasi rollback

Anda dapat CloudFormation memantau status tumpukan Anda selama pembuatan dan pembaruan tumpukan, dan memutar kembali operasi itu jika tumpukan melanggar ambang batas alarm apa pun yang telah Anda tentukan. Tentukan CloudWatch alarm yang CloudFormation harus dipantau. Jika salah satu alarm ALARM berstatus selama operasi tumpukan atau periode pemantauan, putar kembali CloudFormation seluruh operasi tumpukan. Untuk informasi selengkapnya, lihat [Gulung kembali CloudFormation tumpukan Anda pada pelanggaran alarm dengan pemicu rollback](#).

Opsi pemberitahuan

Anda dapat menentukan topik Layanan Pemberitahuan Sederhana Amazon yang baru atau yang sudah ada di mana pemberitahuan tentang peristiwa tumpukan dikirim.

Jika Anda membuat topik Amazon SNS, Anda harus menentukan nama dan alamat email tempat pemberitahuan peristiwa tumpukan akan dikirim.

Opsi pembuatan tumpukan

Opsi berikut disertakan untuk pembuatan tumpukan, tetapi tidak tersedia sebagai bagian dari pembaruan tumpukan.

Batas waktu

Menentukan jumlah waktu, dalam hitungan menit, yang CloudFormation harus mengalokasikan sebelum waktu habis operasi pembuatan tumpukan. Jika tidak CloudFormation dapat membuat seluruh tumpukan dalam waktu yang ditentukan, itu gagal pembuatan tumpukan karena batas waktu dan memutar kembali tumpukan.

Secara default, tidak ada waktu habis untuk pembuatan tumpukan. Namun, sumber daya individu mungkin memiliki waktu habis sendiri berdasarkan sifat layanan yang diterapkan. Misalnya, jika waktu sumber daya individual di tumpukan Anda habis, waktu pembuatan tumpukan juga akan habis meskipun waktu habis yang Anda tentukan untuk pembuatan tumpukan belum tercapai.

Perlindungan penghentian

Mencegah tumpukan tidak sengaja dihapus. Jika pengguna mencoba menghapus tumpukan dengan perlindungan pengakhiran diaktifkan, penghapusan gagal dan tumpukan, termasuk statusnya, tetap tidak berubah. Untuk informasi selengkapnya, lihat [Lindungi CloudFormation tumpukan agar tidak dihapus](#).

Perlindungan pengakhiran Dinonaktifkan secara default.

Pratinjau konfigurasi tumpukan Anda

Untuk melihat bagaimana CloudFormation tumpukan akan dikonfigurasi sebelum membuat tumpukan, buat set perubahan. Fungsi ini memungkinkan Anda untuk memeriksa beragam konfigurasi dan membuat koreksi dan perubahan pada tumpukan Anda sebelum menjalankan set perubahan. Untuk informasi lebih lanjut tentang set perubahan, lihat [Perbarui CloudFormation tumpukan menggunakan set perubahan](#).

Pembuatan set perubahan untuk stack baru

Untuk membuat set perubahan untuk tumpukan baru, pilih templat tumpukan Anda dan tentukan konfigurasi tumpukan seperti yang Anda lakukan jika membuat tumpukan baru, lalu pilih untuk membuat set perubahan baru daripada tumpukan baru.

Untuk membuat set perubahan untuk tumpukan baru

1. Pada halaman Tinjau dan buat, pilih Buat set perubahan.
2. Dalam Buat set perubahan kotak dialog, masukkan nama untuk set perubahan, dan deskripsi jika diinginkan. Pilih Buat set perubahan.

Saat Anda membuat set perubahan untuk tumpukan baru, CloudFormation lakukan hal berikut:

- Meluncurkan tumpukan baru dengan status `REVIEW_IN_PROGRESS`.
- Membuat set perubahan untuk tumpukan baru yang mencerminkan konfigurasi tumpukan yang Anda tentukan di langkah sebelumnya.

CloudFormation menampilkan halaman Ubah set untuk tumpukan yang diusulkan. Saat CloudFormation membuat set perubahan, statusnya adalah `CREATE_IN_PROGRESS`, dan status eksekusinya adalah `UNAVAILABLE`. Ketika CloudFormation selesai berhasil membuat set perubahan, ia menetapkan status set perubahan ke `CREATE_COMPLETE`, dan status eksekusi adalah `AVAILABLE`. Status tumpukan diperbarui ke `REVIEW_IN_PROGRESS`. Pada titik ini, Anda dapat menjalankan set perubahan untuk menyelesaikan pembuatan tumpukan baru.

Di panel Perubahan, CloudFormation menampilkan konfigurasi yang diusulkan dari tumpukan Anda.

Jika CloudFormation gagal membuat set perubahan, itu akan menetapkan status set perubahan ke `CREATE_FAILED`. Perbaiki kesalahan yang ditampilkan di bidang Alasan status, lalu buat set perubahan baru. Pada tahap ini, Anda dapat mencoba berbagai konfigurasi dan membuat koreksi dan perubahan pada tumpukan Anda sebelum menjalankan set perubahan selanjutnya.

3. Untuk menyelesaikan pembuatan tumpukan baru berdasarkan set perubahan, pilih Execute, tentukan konfigurasi rollback Anda, lalu pilih Execute change set.

Lihat informasi tumpukan dari CloudFormation konsol

Setelah Anda membuat CloudFormation tumpukan, Anda dapat menggunakan AWS Management Console untuk melihat data dan sumber dayanya.

Untuk melihat informasi tentang CloudFormation tumpukan Anda

1. Masuk ke AWS Management Console dan buka AWS CloudFormation konsol di <https://console.aws.amazon.com/cloudformation>.
2. Pada bilah navigasi di bagian atas layar, pilih Wilayah AWS tempat tumpukan berada.
3. Pada halaman Stacks, pilih nama tumpukan. CloudFormation menampilkan detail tumpukan untuk tumpukan yang dipilih.

Note

Jika tumpukan telah dihapus, Anda dapat menemukannya dengan menggunakan opsi status Filter. Untuk informasi selengkapnya, lihat [Lihat tumpukan yang dihapus dari konsol CloudFormation](#).

Anda dapat melihat informasi tumpukan berikut:

Info tumpukan

Menampilkan informasi umum tentang tumpukan dan konfigurasinya, termasuk:

Gambaran Umum

Menampilkan nama tumpukan, ID tumpukan, tumpukan root, dan [peran IAM](#), bersama dengan informasi status seperti status tumpukan, status drift, dan perlindungan terminasi.

Tanda

Menampilkan tag yang terkait dengan tumpukan.

Kebijakan tumpukan

Menjelaskan sumber daya tumpukan yang dilindungi terhadap pembaruan tumpukan. Agar Anda dapat memperbarui sumber daya ini, mereka harus secara eksplisit diizinkan selama pembaruan tumpukan. Untuk informasi selengkapnya, lihat [Mencegah pembaruan ke sumber daya tumpukan](#).

Konfigurasi rollback

Menampilkan CloudWatch alarm apa pun yang telah Anda tentukan yang CloudFormation harus dipantau selama operasi tumpukan atau periode pemantauan yang ditentukan. Jika salah satu alarm ALARM berstatus selama operasi tumpukan atau periode pemantauan, putar kembali CloudFormation seluruh operasi tumpukan. Untuk informasi selengkapnya, lihat [Gulung kembali CloudFormation tumpukan Anda pada pelanggaran alarm dengan pemicu rollback](#).

Opsi notifikasi

Menampilkan topik Layanan Pemberitahuan Sederhana Amazon tempat pemberitahuan tentang peristiwa tumpukan dikirim, jika ditentukan.

Peristiwa

Menampilkan operasi yang dilacak saat Anda membuat, memperbarui, atau menghapus tumpukan. Untuk informasi selengkapnya, lihat [Pantau kemajuan tumpukan](#).

Semua kejadian yang dipicu oleh operasi tumpukan tertentu ditetapkan dengan token permintaan klien yang sama, yang dapat Anda gunakan untuk melacak operasi. Operasi tumpukan yang dimulai dari konsol menggunakan format token Console- StackOperation -ID, yang membantu Anda mengidentifikasi operasi tumpukan dengan mudah. Misalnya, jika Anda membuat tumpukan menggunakan konsol, setiap kejadian tumpukan yang dihasilkan akan ditetapkan token yang sama dalam format berikut: Console-CreateStack-7f59c3cf-00d2-40c7-b2ff-e75db0987002.

Sumber daya

Menampilkan sumber daya yang merupakan bagian dari tumpukan.

Output

Menampilkan output yang dideklarasikan dalam templat tumpukan. Untuk informasi selengkapnya, lihat [Dapatkan output yang diekspor dari tumpukan yang diterapkan CloudFormation](#).

Parameter

Menampilkan parameter tumpukan dan nilai mereka.

Untuk tumpukan yang berisi parameter Systems Manager, kolom Nilai Terselesaikan menampilkan nilai yang digunakan dalam definisi tumpukan untuk parameter Systems Manager.

Untuk informasi selengkapnya, lihat [Tentukan sumber daya yang ada saat runtime dengan CloudFormation tipe parameter yang disediakan](#).

Templat

Menampilkan templat tumpukan.

Untuk tumpukan yang berisi makro, pilih Lihat templat asli untuk melihat templat yang dikirimkan pengguna, atau Lihat templat yang diproses untuk melihat templat setelah CloudFormation memproses makro yang direferensikan. CloudFormation menggunakan template yang diproses untuk membuat atau memperbarui tumpukan Anda.

Ubah set

Menampilkan set perubahan tumpukan.

Untuk informasi selengkapnya, lihat [Melihat set perubahan untuk CloudFormation tumpukan](#).

Sinkronisasi Git

Menampilkan dasbor sinkronisasi Git stack.

Untuk informasi selengkapnya, lihat [Dasbor status sinkronisasi Git](#).

Perbarui template tumpukan Anda

Untuk memodifikasi sumber daya atau properti dalam CloudFormation tumpukan, Anda harus memperbarui template tumpukan. Mulailah dengan template yang ada untuk tumpukan itu dan buat perubahan Anda padanya. Jika Anda memiliki template yang disimpan dalam sistem kontrol sumber, gunakan salinannya sebagai titik awal Anda. Jika tidak, Anda bisa mendapatkan salinan template dari CloudFormation.

Jika Anda hanya ingin mengubah parameter atau pengaturan tumpukan (seperti topik Amazon SNS tumpukan), Anda dapat menggunakan kembali template yang ada tanpa mendapatkan salinannya.

Anda dapat memperbarui template CloudFormation tumpukan dengan menggunakan editor teks atau [Komposer Infrastruktur](#).

Untuk memperbarui template stack yang ada dengan menggunakan Infrastructure Composer

1. Masuk ke AWS Management Console dan buka AWS CloudFormation konsol di <https://console.aws.amazon.com/cloudformation>.
2. Pada halaman Stacks, pilih nama tumpukan yang akan diperbarui.

3. Pilih tab Template, lalu pilih View in Infrastructure Composer.

CloudFormation membuka template di Infrastructure Composer.

4. Perbarui template Anda menggunakan salah satu metode berikut:

- Antarmuka kanvas: Di sini, Anda dapat menarik dan melepas dari palet Resources. Konfigurasi sumber daya dengan mengklik dua kali pada kartu untuk membuka panel Resource properties. Connect sumber daya sesuai kebutuhan. Untuk petunjuk mendetail tentang penggunaan antarmuka Canvas, lihat [Cara menulis](#). AWS Infrastructure Composer
- Antarmuka template: Beralih dari Canvas ke antarmuka Template. Buat pembaruan in-line ke kode template. Beralih antara format JSON ke YAML sesuai kebutuhan.

5. Pilih Validasi untuk memeriksa kesalahan sintaks dalam template.

6. Ketika Anda siap untuk mengeksport perubahan ke CloudFormation, pilih Perbarui template.

Untuk memperbarui template tumpukan yang ada dengan menggunakan AWS CLI

1. Untuk mendapatkan template untuk tumpukan yang ingin Anda perbarui, gunakan perintah [CLI get-template](#).
2. Salin templat, tempelkan ke file teks, ubah, dan simpan. Salin hanya templat. Perintah membungkus templat dalam tanda kutip, tetapi tidak menyalin tanda kutip di sekitar templat. Templat itu sendiri diawali dengan kurung buka dan diakhiri dengan kurung tutup. Tentukan perubahan pada sumber daya tumpukan dalam file ini.

Ingatlah poin-poin berikut saat Anda membuat perubahan pada template Anda:

- Anda tidak dapat menambahkan, memodifikasi, atau menghapus parameter yang digunakan oleh sumber daya yang tidak mendukung pembaruan.
- Untuk sebagian besar sumber daya, mengubah nama logis dari sumber daya setara dengan menghapus sumber daya tersebut dan menggantinya dengan yang baru. Sumber daya lainnya yang bergantung pada sumber daya yang diganti namanya juga perlu diperbarui dan dapat menyebabkan mereka diganti. Sumber daya lain mengharuskan Anda untuk memperbarui properti (bukan hanya nama logis) untuk memulai pembaruan.
- Beberapa sumber daya mungkin memiliki kendala tentang nilai apa yang dapat Anda tetapkan untuk properti tertentu. Misalnya, perubahan `AllocatedStorage` properti untuk instance database RDS harus lebih besar dari nilai saat ini. Jika pembaruan Anda melanggar aturan ini, bagian itu akan gagal.

- Memperbarui satu sumber daya juga dapat memengaruhi orang lain yang mereferensikannya. Jika Anda menggunakan fungsi seperti [Fungsi Ref](#) atau [Fungsi Fn::GetAtt](#) untuk mengatur properti berdasarkan sumber daya lain, CloudFormation akan memperbarui sumber referensi juga ketika referensi berubah.
- Untuk informasi tentang efek memperbarui properti sumber daya tertentu, lihat [referensi jenis AWS sumber daya dan properti](#). Untuk setiap properti, efek pembaruan akan menjadi salah satu dari berikut:
 - Pembaruan membutuhkan: [Tidak ada gangguan](#)
 - Pembaruan membutuhkan: [Beberapa gangguan](#)
 - Pembaruan membutuhkan: [Penggantian](#)
- Anda dapat memverifikasi sintaks JSON atau YAMAL template Anda dengan menggunakan perintah [validate-template](#) CLI atau dengan menentukan template Anda di konsol. Konsol melakukan validasi secara otomatis. Namun, metode ini hanya memverifikasi sintaks template Anda dan tidak memvalidasi nilai properti yang Anda tentukan untuk sumber daya yang valid untuk sumber daya tersebut. [Untuk validasi yang lebih kompleks atau untuk memeriksa praktik terbaik, Anda mungkin juga menggunakan alat tambahan seperti CloudFormation Linter \(cfn-lint\) dan CloudFormation Rain \(rain fmt\).](#)

Note

Terkadang CloudFormation tidak mengizinkan perubahan tertentu yang Anda coba buat, dan itu akan memberi tahu Anda bahwa perubahan itu tidak diizinkan. Namun, pesan ini mungkin terjadi secara asinkron, karena sumber daya dibuat dan diperbarui oleh CloudFormation dalam urutan non-deterministik secara default.

Memahami perilaku pembaruan sumber daya tumpukan

Saat Anda mengirimkan AWS CloudFormation pembaruan, memperbarui sumber daya berdasarkan perbedaan antara apa yang Anda kirimkan dan templat tumpukan saat ini. Sumber daya yang tidak berubah berjalan tanpa gangguan selama proses pembaruan. Untuk sumber daya yang diperbarui, AWS CloudFormation gunakan salah satu perilaku pembaruan berikut:

Update tanpa gangguan

AWS CloudFormation memperbarui sumber daya tanpa mengganggu pengoperasian sumber daya itu dan tanpa mengubah ID fisik sumber daya. Misalnya, jika Anda memperbarui properti tertentu pada [AWS::CloudTrail::Trail](#) sumber daya, AWS CloudFormation perbarui jejak tanpa gangguan.

Pembaruan dengan Beberapa Gangguan

AWS CloudFormation memperbarui sumber daya dengan beberapa gangguan. Misalnya, jika Anda memperbarui properti tertentu pada [AWS::EC2::Instance](#) sumber daya, instance mungkin mengalami gangguan sementara AWS CloudFormation dan Amazon EC2 mengonfigurasi ulang instance.

Penggantian

AWS CloudFormation membuat ulang sumber daya selama pembaruan, yang juga menghasilkan ID fisik baru. AWS CloudFormation biasanya membuat sumber daya pengganti terlebih dahulu, mengubah referensi dari sumber daya dependen lainnya untuk menunjuk ke sumber daya pengganti, dan kemudian menghapus sumber daya lama. Misalnya, jika Anda memperbarui AvailabilityZone properti tipe [AWS::EC2::Instance](#) sumber daya, AWS CloudFormation membuat sumber daya baru dan mengganti sumber daya EC2 Instance saat ini dengan yang baru.

Jika Anda menambahkan atau menghapus properti yang memerlukan penggantian, itu juga akan memicu pembaruan. Pembaruan akan terjadi bahkan jika nilai sebenarnya dari properti tidak berubah.

Metode yang AWS CloudFormation digunakan tergantung pada properti yang Anda perbarui untuk jenis sumber daya tertentu. Perilaku pembaruan untuk setiap properti dijelaskan dalam [referensi jenis AWS sumber daya dan properti](#).

Tergantung pada perilaku pembaruan, Anda dapat memutuskan kapan untuk memodifikasi sumber daya untuk mengurangi dampak perubahan ini pada aplikasi Anda. Secara khusus, Anda dapat merencanakan kapan sumber daya harus diganti selama pembaruan. Misalnya, jika Anda memperbarui Port properti tipe DBInstance sumber daya [AWS::RDS::](#), AWS CloudFormation ganti instans DB dengan membuat instans DB baru dengan pengaturan port yang diperbarui dan menghapus instans DB lama. Sebelum pembaruan, Anda dapat berencana untuk melakukan hal berikut untuk mempersiapkan penggantian basis data:

- Ambil snapshot basis data saat ini.
- Siapkan strategi untuk cara aplikasi yang menggunakan instans DB akan menangani gangguan saat instans DB sedang diganti.
- Pastikan bahwa aplikasi yang menggunakan instans DB menganggap pengaturan port yang diperbarui dan pembaruan lain yang telah Anda buat.
- Gunakan snapshot DB untuk memulihkan basis data pada instans DB baru.

Contoh ini tidak lengkap, tapi ini dimaksudkan untuk memberi Anda gambaran tentang hal-hal yang harus direncanakan ketika sumber daya diganti selama pembaruan.

Note

Jika template menyertakan satu atau beberapa [tumpukan bersarang](#), AWS CloudFormation juga memulai pembaruan untuk setiap tumpukan bersarang. Ini diperlukan untuk menentukan apakah tumpukan bersarang telah dimodifikasi. AWS CloudFormation hanya memperbarui sumber daya tersebut di tumpukan bersarang yang memiliki perubahan yang ditentukan dalam templat yang sesuai.

Perbarui CloudFormation tumpukan menggunakan set perubahan

Saat Anda perlu memperbarui tumpukan, memahami bagaimana perubahan Anda akan memengaruhi sumber daya yang berjalan sebelum Anda menerapkannya dapat membantu Anda memperbarui tumpukan dengan percaya diri. Set perubahan memungkinkan Anda melihat pratinjau bagaimana perubahan yang diusulkan pada tumpukan dapat memengaruhi sumber daya yang sedang berjalan, termasuk dampaknya pada properti dan atribut sumber daya. Apakah perubahan Anda akan menghapus atau mengganti sumber daya penting apa pun, CloudFormation membuat perubahan pada tumpukan Anda hanya ketika Anda memutuskan untuk menjalankan set perubahan, memungkinkan Anda untuk memutuskan apakah akan melanjutkan perubahan yang Anda usulkan atau menjelajahi perubahan lain dengan membuat set perubahan lain. Anda dapat membuat dan mengelola set perubahan menggunakan CloudFormation konsol, AWS CLI, atau CloudFormation API.

Topik

- [Buat set perubahan untuk CloudFormation tumpukan](#)
- [Melihat set perubahan untuk CloudFormation tumpukan](#)

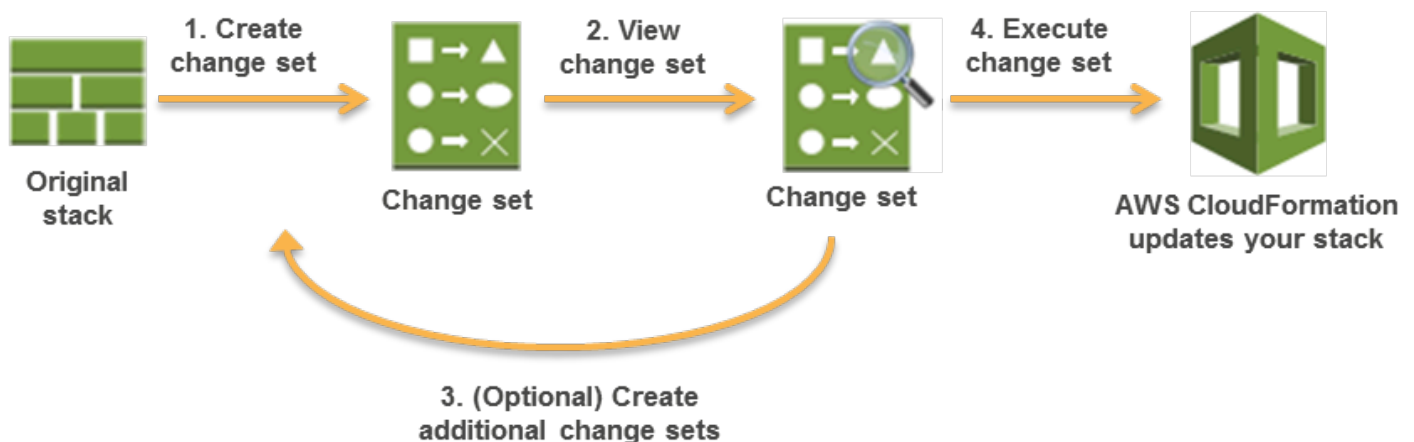
- [Jalankan set perubahan untuk CloudFormation tumpukan](#)
- [Menghapus set perubahan untuk CloudFormation tumpukan](#)
- [Contoh set perubahan untuk CloudFormation tumpukan](#)
- [Set perubahan untuk tumpukan yang di-nest](#)

⚠ Important

Ubah set tidak menunjukkan apakah CloudFormation akan berhasil memperbarui tumpukan. Misalnya, set perubahan tidak memeriksa apakah Anda akan melampaui kuota akun, jika Anda memperbarui sumber daya yang tidak mendukung pembaruan, atau jika Anda memiliki izin yang tidak memadai untuk mengubah sumber daya, yang semuanya dapat menyebabkan pembaruan tumpukan gagal. Jika pembaruan gagal, CloudFormation mencoba untuk roll back sumber daya Anda ke status semula.

Ubah Gambaran Umum Set

Diagram berikut merangkum bagaimana Anda menggunakan set perubahan untuk memperbarui tumpukan:



1. Buat set perubahan dengan mengirimkan perubahan untuk tumpukan yang ingin Anda perbarui. Anda dapat mengirimkan template tumpukan yang dimodifikasi atau nilai parameter input yang dimodifikasi. CloudFormation membandingkan tumpukan Anda dengan perubahan yang Anda kirimkan untuk menghasilkan set perubahan; itu tidak membuat perubahan pada tumpukan Anda pada saat ini.

2. Lihat set perubahan untuk melihat pengaturan tumpukan dan sumber daya mana yang akan berubah. Misalnya, Anda dapat melihat sumber daya mana yang CloudFormation akan ditambahkan, dimodifikasi, atau dihapus. Selain itu, Anda dapat melihat before-and-after perbandingan properti sumber daya dan atribut, seperti tag, yang CloudFormation akan berubah.
3. Opsional: Jika Anda ingin mempertimbangkan perubahan lain sebelum Anda memutuskan perubahan mana yang akan dibuat, buat set perubahan tambahan. Membuat beberapa set perubahan membantu Anda memahami dan mengevaluasi bagaimana perubahan yang berbeda akan memengaruhi sumber daya dan properti Anda. Anda dapat membuat set perubahan sebanyak yang Anda perlukan.
4. Jalankan set perubahan yang berisi perubahan yang ingin Anda terapkan ke tumpukan Anda. CloudFormation memperbarui tumpukan Anda dengan perubahan tersebut.

Note

Setelah Anda menjalankan perubahan, CloudFormation menghapus semua set perubahan yang terkait dengan tumpukan karena tidak berlaku untuk tumpukan yang diperbarui.

Anda juga dapat menghapus set perubahan untuk mencegah menjalankan set perubahan yang seharusnya tidak diterapkan.

Buat set perubahan untuk CloudFormation tumpukan

Untuk membuat set perubahan untuk tumpukan yang sedang berjalan, kirimkan perubahan yang ingin Anda buat dengan menyediakan templat yang dimodifikasi, nilai parameter input baru, atau keduanya. CloudFormation menghasilkan set perubahan dengan membandingkan tumpukan Anda dengan perubahan yang Anda kirimkan.

Anda dapat memodifikasi templat sebelum membuat set perubahan atau selama pembuatan set perubahan.

Create a change set (console)

Untuk membuat set perubahan

1. Buka AWS CloudFormation konsol di <https://console.aws.amazon.com/cloudformation>.
2. Pada bilah navigasi di bagian atas layar, pilih Anda Wilayah AWS.
3. Pada halaman Stacks, pilih tumpukan berjalan yang ingin Anda buat set perubahan.

4. Di panel detail tumpukan, pilih Tindakan tumpukan, lalu pilih Buat set perubahan.
5. Pada Buat perubahan set untuk **stack-name** halaman, lakukan salah satu hal berikut untuk mengubah nilai parameter input, menentukan lokasi template yang diperbarui, atau memodifikasi template:

Tugas	Tindakan
Untuk memodifikasi nilai parameter input	Pilih Gunakan templat yang ada, lalu pilih Berikutnya untuk melanjutkan untuk memasukkan atau memodifikasi nilai parameter input.
Untuk menentukan lokasi templat yang diperbarui	Jika Anda telah memodifikasi template, pilih Ganti template yang ada, lalu lakukan salah satu hal berikut: • Untuk template yang disimpan dalam bucket Amazon S3, pilih URL Amazon S3. Masukkan atau tempel URL untuk templat, lalu pilih Selanjutnya. Jika Anda memiliki template dalam bucket berkemampuan versi, Anda dapat menentukan versi template tertentu dengan menambahkan <code>?versionId= version-id</code> ke URL. Untuk informasi selengkapnya, lihat Bekerja dengan objek dalam bucket berkemampuan versi di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon. • Untuk templat yang disimpan secara lokal di komputer Anda, pilih Unggah file templat. Pilih Pilih File untuk menavigasi ke file dan memilihnya, dan kemudian pilih Selanjutnya.
Untuk memodifikasi templat	Jika Anda belum memodifikasi template, pilih Edit template di Infrastructure Composer, lalu pilih Edit di Infrastructure Composer. Anda dialihkan ke AWS Infrastructure Composer. Setelah Anda memodifikasi template, pilih Buat set perubahan dan kemudian Konfirmasi dan terus kembali CloudFormation ke Buat perubahan set untuk stack-name halaman, lalu pilih Berikutnya.

6. Pada halaman Tentukan detail tumpukan, tentukan nama untuk set perubahan dan secara opsional tentukan deskripsi set perubahan untuk mengidentifikasi tujuannya di bagian

Ikhtisar. Jika templat anda berisi parameter, pada halaman Tentukan detail tumpukan, masukkan atau modifikasi nilai parameter input yang berlaku, dan kemudian pilih Selanjutnya.

Jika Anda menggunakan kembali templat tumpukan, CloudFormation isi setiap parameter dengan nilai saat ini di tumpukan, dengan pengecualian parameter yang dideklarasikan dengan atribut. NoEcho Untuk menggunakan nilai yang ada untuk parameter tersebut, pilihGunakan nilai yang ada.

Untuk informasi selengkapnya tentang penggunaan NoEcho untuk menutupi informasi sensitif, dan menggunakan parameter dinamis untuk mengelola rahasia, lihat praktik [Jangan menanamkan kredensial dalam templat Anda](#) terbaik.

7. Pada halaman Configure stack options, perbarui tag stack, peran layanan IAM, kebijakan tumpukan, konfigurasi rollback, topik notifikasi Amazon SNS (jika ada), atau ubah set.

 Note

Set perubahan untuk tumpukan nest Diaktifkan secara default, yang akan membuat set perubahan untuk semua tumpukan nest yang ditentukan di templat Anda. Untuk membuat set perubahan untuk tumpukan saat ini saja, pilihNonaktifkan. Untuk informasi selengkapnya tentang set perubahan untuk tumpukan bersarang, lihat. [Set perubahan untuk tumpukan yang di-nest](#)

8. Jika template menyertakan sumber daya IAM, untuk Kemampuan, pilih Saya mengakui yang CloudFormation mungkin membuat sumber daya IAM. Sumber daya IAM dapat mengubah izin di AWS akun Anda; tinjau sumber daya ini untuk memastikan bahwa Anda hanya mengizinkan tindakan yang Anda inginkan. Untuk informasi selengkapnya, lihat [Mengakui sumber daya IAM dalam templat CloudFormation](#).
9. Pilih Next untuk melanjutkan.
10. Pada *stack-name* halaman Tinjauan, tinjau perubahan untuk set perubahan ini.
11. Pilih Kirim.

Anda dialihkan ke tab Perubahan di halaman detail set perubahan. Saat CloudFormation menghasilkan set perubahan, status set perubahan adalahCREATE_PENDING. Setelah membuat set perubahan, CloudFormation atur statusnya menjadiCREATE_COMPLETE. Di bagian Perubahan, CloudFormation daftar semua perubahan yang akan dilakukan pada tumpukan Anda. Untuk informasi selengkapnya, lihat [Melihat set perubahan untuk CloudFormation tumpukan](#).

Pilih Lihat detail di kolom Perubahan tingkat properti untuk melihat perubahan yang dibuat di tingkat properti.

Jika CloudFormation gagal membuat set perubahan (FAILEDstatus laporan), perbaiki kesalahan yang ditampilkan di bidang Status, lalu buat ulang set perubahan.

12. Setelah mengonfirmasi perubahan terlihat benar, pilih Jalankan set perubahan

Create a change set for nested stacks (console)

Untuk membuat set perubahan untuk tumpukan bersarang

1. Buka AWS CloudFormation konsol di <https://console.aws.amazon.com/cloudformation>.
2. Pada bilah navigasi di bagian atas layar, pilih Anda Wilayah AWS.
3. Pada halaman Stacks, pilih tumpukan berjalan yang ingin Anda buat set perubahan.
4. Di panel detail tumpukan, pilih Tindakan tumpukan, lalu pilih Buat set perubahan.
5. Pada Buat perubahan set untuk **stack-name** halaman, lakukan salah satu hal berikut untuk mengubah nilai parameter input, menentukan lokasi template yang diperbarui, atau memodifikasi template:

Tugas	Tindakan
Untuk memodifikasi nilai parameter input	Pilih Gunakan templat yang ada, lalu pilih Berikutnya untuk melanjutkan untuk memasukkan atau memodifikasi nilai parameter input.
Untuk menentukan lokasi templat yang diperbarui	Jika Anda telah memodifikasi template, pilih Ganti template yang ada, lalu lakukan salah satu hal berikut: • Untuk template yang disimpan dalam bucket Amazon S3, pilih URL Amazon S3. Masukkan atau tempel URL untuk templat, lalu pilih Selanjutnya. Jika Anda memiliki template dalam bucket berkemampuan versi, Anda dapat menentukan versi template tertentu dengan menambahkan ?versionId= version-id ke URL. Untuk informasi selengkapnya, lihat Bekerja dengan objek dalam

Tugas	Tindakan
	bucket berkemampuan versi di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon. • Untuk templat yang disimpan secara lokal di komputer Anda, pilih Unggah file templat. Pilih Pilih File untuk menavigasi ke file dan memilihnya, dan kemudian pilih Selanjutnya.
Untuk memodifikasi templat	Jika Anda belum memodifikasi template, pilih Edit template di Infrastructure Composer, lalu pilih Edit di Infrastructure Composer. Anda dialihkan ke AWS Infrastructure Composer. Setelah Anda memodifikasi template, pilih Buat set perubahan dan kemudian Konfirmasi dan terus kembali CloudFormation ke Buat perubahan set untuk stack-name halaman, lalu pilih Berikutnya.

6. Pada halaman Tentukan detail tumpukan, tentukan nama untuk set perubahan dan secara opsional tentukan deskripsi set perubahan untuk mengidentifikasi tujuannya di bagian Ikhtisar. Jika templat anda berisi parameter, pada halaman Tentukan detail tumpukan, masukkan atau modifikasi nilai parameter input yang berlaku, dan kemudian pilih Selanjutnya.

Jika Anda menggunakan kembali templat tumpukan, CloudFormation isi setiap parameter dengan nilai saat ini di tumpukan, dengan pengecualian parameter yang dideklarasikan dengan atribut. NoEcho Untuk menggunakan nilai yang ada untuk parameter tersebut, pilih Gunakan nilai yang ada.

Untuk informasi selengkapnya tentang penggunaan NoEcho untuk menutupi informasi sensitif, serta menggunakan parameter dinamis untuk mengelola rahasia, lihat praktik [Jangan menanamkan kredensial dalam templat Anda](#) terbaik.

7. Pada halaman Configure stack options, perbarui tag stack, peran layanan IAM, kebijakan tumpukan, konfigurasi rollback, topik notifikasi Amazon SNS (jika ada), atau ubah set. Untuk informasi selengkapnya, lihat [Konfigurasi opsi tumpukan](#).

Note

Set perubahan untuk tumpukan nest Diaktifkan secara default, yang akan membuat set perubahan untuk semua tumpukan nest yang ditentukan di templat Anda. Untuk

informasi selengkapnya tentang set perubahan untuk tumpukan bersarang, lihat [Set perubahan untuk tumpukan yang di-nest](#)

8. Jika template menyertakan sumber daya IAM, untuk Kemampuan, pilih Saya mengakui yang CloudFormation mungkin membuat sumber daya IAM. Sumber daya IAM dapat mengubah izin di AWS akun Anda; tinjau sumber daya ini untuk memastikan bahwa Anda hanya mengizinkan tindakan yang Anda inginkan. Untuk informasi selengkapnya, lihat [Mengakui sumber daya IAM dalam templat CloudFormation](#).
9. Pilih Next untuk melanjutkan.
10. Pada *stack-name* halaman Tinjauan, tinjau perubahan untuk set perubahan ini.
11. Pilih Kirim.

 Note

CloudFormation set perubahan tingkat properti tidak menyelesaikan referensi lintas-tumpukan saat Anda membuat set perubahan untuk tumpukan bersarang. Set perubahan dapat menandai sumber daya dalam tumpukan anak untuk penggantian bersyarat jika mereka mereferensikan output dari tumpukan induk, dan tumpukan induk telah dimodifikasi

Anda dialihkan ke tab Perubahan di halaman detail set perubahan. Saat CloudFormation menghasilkan set perubahan, status set perubahan adalah `CREATE_PENDING`. Setelah membuat set perubahan, CloudFormation atur statusnya menjadi `CREATE_COMPLETE`. Di bagian Perubahan, CloudFormation daftar semua perubahan yang akan dilakukan pada tumpukan Anda. Untuk informasi selengkapnya, lihat [Melihat set perubahan untuk CloudFormation tumpukan](#).

Jika CloudFormation gagal membuat set perubahan (`FAILED` status laporan), perbaiki kesalahan yang ditampilkan di bidang Status, lalu buat ulang set perubahan.

12. Setelah mengonfirmasi perubahan terlihat benar, pilih Jalankan set perubahan

Untuk membuat set perubahan (AWS CLI)

- Gunakan perintah [create-change-set](#).

Anda mengirimkan perubahan Anda sebagai opsi perintah. Anda dapat menentukan nilai parameter baru, templat yang dimodifikasi, atau keduanya. Misalnya, perintah berikut membuat set perubahan bernama `SampleChangeSet` untuk tumpukan `MyStack`. Set perubahan menggunakan templat tumpukan saat ini, tetapi dengan nilai yang berbeda untuk parameter `Purpose`:

```
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"
```

Melihat set perubahan untuk CloudFormation tumpukan

Setelah Anda membuat set perubahan, Anda dapat melihat perubahan yang diusulkan sebelum menjalankannya. Anda dapat menggunakan CloudFormation konsol, AWS CLI, atau CloudFormation API untuk melihat set perubahan. Konsol CloudFormation memberikan ringkasan dari perubahan dan daftar perubahan terperinci dalam format JSON. AWS CloudFormation API AWS CLI dan mengembalikan daftar perubahan terperinci dalam format JSON.

View a change set (console)

Untuk melihat set perubahan

1. Buka AWS CloudFormation konsol di <https://console.aws.amazon.com/cloudformation>.
2. Pada bilah navigasi di bagian atas layar, pilih Anda Wilayah AWS.
3. Pada halaman Stacks, pilih nama tumpukan yang berisi set perubahan yang ingin Anda lihat.
4. Di panel navigasi, pilih Set PerubahanUntuk melihat daftar set perubahan tumpukan.
5. Pilih nama set perubahan yang ingin Anda lihat.

CloudFormation Konsol mengarahkan Anda ke halaman detail set perubahan, di mana Anda dapat melihat waktu set perubahan dibuat, statusnya, input yang digunakan untuk menghasilkan set perubahan, dan ringkasan perubahan.

Di bagian Ubah, setiap baris mewakili sumber daya yang akan ditambahkan, diubah, atau dihapus oleh CloudFormation .

- Tambah - CloudFormation membuat sumber daya saat Anda menambahkan sumber daya ke template tumpukan.
- Modify — CloudFormation memodifikasi sumber daya saat Anda mengubah properti sumber daya di template stack.
- Hapus — CloudFormation menghapus sumber daya saat Anda menghapus sumber daya dari template tumpukan.

 Note

Modifikasi dapat menyebabkan sumber daya terganggu atau diganti (dibuat kembali). Untuk informasi lebih lanjut tentang perilaku pembaruan sumber daya, lihat [Memahami perilaku pembaruan sumber daya tumpukan](#).

Untuk fokus pada perubahan tertentu, gunakan tampilan filter. Misalnya, filter untuk tipe sumber daya tertentu, seperti `AWS::EC2::Instance`. Untuk memfilter sumber daya tertentu, tentukan ID logis atau fisik, seperti `myWebServer` atau `i-123abcd4`.

6. Di bagian Perubahan, pilih Lihat detail di kolom Perubahan tingkat properti untuk melihat perubahan nilai properti yang dibuat pada sumber daya Anda.
7. CloudFormation Konsol mengarahkan Anda ke halaman perubahan tingkat properti untuk sumber daya, di mana Anda dapat melihat konfigurasi templat sumber daya sebelum menjalankan set perubahan dan seperti apa konfigurasi templat setelah menjalankan set perubahan.

Tabel bagian perubahan tingkat properti menunjukkan nilai Path, Change type, Before value, dan After untuk properti yang terpengaruh. Dalam tabel, pilih kotak centang untuk setiap perubahan yang ingin Anda sorot dalam tampilan Sebelum dan Setelah template Anda untuk melihat perubahan apa yang akan dilakukan di tingkat properti.

- Tambah - Properti yang ditambahkan disorot hijau.
- Modifikasi - Properti yang dimodifikasi disorot biru.
- Hapus - Properti yang dihapus disorot merah.

View a change set for nested stack (console)

Untuk melihat set perubahan untuk tumpukan nest (konsol)

1. Buka AWS CloudFormation konsol di <https://console.aws.amazon.com/cloudformation>.
2. Pada bilah navigasi di bagian atas layar, pilih Anda Wilayah AWS.
3. Pada halaman Stacks, pilih nama tumpukan yang berisi set perubahan yang ingin Anda lihat.
4. Di panel navigasi, pilih Ubah set untuk melihat daftar set perubahan tumpukan.
5. Pilih nama set perubahan yang ingin Anda lihat.

CloudFormation Konsol mengarahkan Anda ke halaman detail set perubahan, di mana Anda dapat melihat waktu set perubahan dibuat, statusnya, input yang digunakan untuk menghasilkan set perubahan, dan ringkasan perubahan.

Di bagian Ubah, setiap baris mewakili sumber daya yang akan ditambahkan, dimodifikasi, dihapus, atau ditampilkan status dinamis oleh CloudFormation .

- Tambah - CloudFormation membuat sumber daya saat Anda menambahkan sumber daya ke template tumpukan.
- Modify — CloudFormation memodifikasi sumber daya saat Anda mengubah properti sumber daya di template stack.
- Hapus — CloudFormation menghapus sumber daya saat Anda menghapus sumber daya dari template tumpukan.
- Dinamis - tidak CloudFormation dapat menentukan tindakan perubahan sumber daya yang tepat dari template tumpukan bersarang.

Note

Modifikasi dapat menyebabkan sumber daya terganggu atau diganti (dibuat kembali). Untuk informasi lebih lanjut tentang perilaku pembaruan sumber daya, lihat [Memahami perilaku pembaruan sumber daya tumpukan](#).

Untuk fokus pada perubahan tertentu, gunakan tampilan filter. Misalnya, filter untuk tipe sumber daya tertentu, seperti **AWS::CloudFormation::Stack**. Untuk memfilter sumber daya tertentu, tentukan ID logis atau fisik, seperti **DeadLetterQueue** atau **NestedStack**.

6. Di bagian Ubah, pilih Lihat set perubahan nest pada set perubahan nestt yang ingin Anda lihat.

CloudFormation Konsol mengarahkan Anda ke halaman detail set perubahan bersarang. Anda dapat memilih Masuk ke set perubahan akar untuk melihat set perubahan akar atau, Anda dapat memilih Lihat set perubahan induk untuk melihat set perubahan induk. Untuk informasi lebih lanjut lihat, [Set perubahan untuk tumpukan yang di-nest](#).

 Note

CloudFormation set perubahan tingkat properti tidak menyelesaikan referensi cross-stack saat Anda membuat set perubahan untuk tumpukan bersarang. Set perubahan dapat menandai sumber daya dalam tumpukan anak untuk penggantian bersyarat jika mereka mereferensikan output dari tumpukan induk, dan tumpukan induk telah dimodifikasi

Untuk melihat set perubahan (AWS CLI)

1. Untuk mendapatkan ID dari set perubahan, jalankan [change-sets](#) perintah.

Tentukan nama tumpukan yang memiliki set perubahan yang ingin Anda lihat, seperti yang ditunjukkan pada contoh berikut:

```
aws cloudformation list-change-sets --stack-name MyStack
```

CloudFormation mengembalikan daftar set perubahan, mirip dengan berikut:

```
{
  "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. Jalankan [describe-change-set](#) perintah, menentukan ID dari set perubahan yang ingin Anda lihat. Sebagai contoh:

```
aws cloudformation describe-change-set \
  --change-set-name arn:aws:cloudformation:us-east-1:123456789012:changeSet/SampleChangeSet/1a2345b6-0000-00a0-a123-00abc0abc000
```

CloudFormation mengembalikan informasi tentang set perubahan yang ditentukan.

```
{
  "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"
}
```

Gunakan `--include-property-values` dengan `describe-change-set` untuk membuat daftar perubahan tingkat properti.

Perubahan daftar kunci Changes ke sumber daya. Jika Anda menjalankan set perubahan ini, CloudFormation akan memperbarui tag `i-1abc23d4` EC2 instance. Untuk deskripsi setiap bidang, lihat tipe [Change](#) data di Referensi AWS CloudFormation API.

Untuk contoh tambahan set perubahan, lihat [Contoh set perubahan untuk CloudFormation tumpukan](#).

Untuk melihat perubahan tingkat properti dalam set perubahan ()AWS CLI

- Perintah berikut mencantumkan perubahan tingkat properti yang terkait dengan set perubahan untuk `AWS::EC2::NetworkInterface` sumber daya yang akan menghapus `Ipv4Prefixes` properti, memodifikasi sumber daya, dan menambahkan: `Description Tag`

```
aws cloudformation describe-change-set --include-property-values \  
  --change-set-name arn:aws:cloudformation:us-east-1:123456789012:changeSet/  
ExampleChangeSet/9f7b541b-126b-44f7-998e-932174557841
```

Berikut ini adalah output contoh.

```
"ChangeSetName": "ExampleChangeSet",  
  "ChangeSetId": "arn:aws:cloudformation:us-east-1:803642222207:changeSet/  
ExampleChangeSet/9f7b541b-126b-44f7-998e-932174557841",  
  "StackId": "arn:aws:cloudformation:us-east-1:803642222207:stack/ExampleStack/  
ab664180-f686-11ee-9e29-12cd92393671",  
  "StackName": "ExampleStack",  
  "Description": null,  
  "Parameters": null,  
  "CreationTime": "2024-04-09T18:04:59.935000+00:00",  
  "ExecutionStatus": "AVAILABLE",  
  "Status": "CREATE_COMPLETE",  
  "StatusReason": null,  
  "NotificationARNs": [],  
  "RollbackConfiguration": {  
    "RollbackTriggers": []  
  },  
  "Capabilities": [],  
  "Tags": null,
```

```

    "ParentChangeSetId": null,
    "IncludeNestedStacks": true,
    "RootChangeSetId": null,
    "OnStackFailure": null,
  {
    "Changes": [
      {
        "Type": "Resource",
        "ResourceChange": {
          "Action": "Modify",
          "LogicalResourceId":
"EC2NetworkInterface00eni067fd35b649a05b7100Tpyls",
          "PhysicalResourceId": "eni-067fd35b649a05b71",
          "ResourceType": "AWS::EC2::NetworkInterface",
          "Replacement": "False",
          "Scope": [
            "Properties",
            "Tags"
          ],
          "Details": [
            {
              "Target": {
                "Attribute": "Properties",
                "Name": "Ipv4Prefixes",
                "RequiresRecreation": "Never",
                "Path": "/Properties/Ipv4Prefixes",
                "BeforeValue": "[]",
                "AttributeChangeType": "Remove"
              },
              "Evaluation": "Static",
              "ChangeSource": "DirectModification"
            },
            {
              "Target": {
                "Attribute": "Properties",
                "Name": "Description",
                "RequiresRecreation": "Never",
                "Path": "/Properties/Description",
                "BeforeValue": "",
                "AfterValue": "Description",
                "AttributeChangeType": "Modify"
              },
              "Evaluation": "Static",
              "ChangeSource": "DirectModification"
            }
          ]
        }
      }
    ]
  }

```



```

    },
    {
      "Target": {
        "Attribute": "Tags",
        "RequiresRecreation": "Never",
        "Path": "/Properties/Tags/0",
        "AfterValue": "{\"Key\":\"Test\",\"Value\":\"Test\"}",
        "AttributeChangeType": "Add"
      },
      "Evaluation": "Static",
      "ChangeSource": "DirectModification"
    }
  ],
  "BeforeContext": "{\"Properties\":{\"Description\":\"\", \"PrivateIpAddress\":\"172.31.76.2\", \"PrivateIpAddresses\": [{\"PrivateIpAddress\": \"172.31.76.2\", \"Primary\": \"true\"}], \"SecondaryPrivateIpAddressCount\": \"0\", \"Ipv6PrefixCount\": \"0\", \"Ipv4Prefixes\": [], \"Ipv4PrefixCount\": \"0\", \"GroupSet\": [\"sg-05a45689b1059e82d\"], \"Ipv6Prefixes\": [], \"SubnetId\": \"subnet-455e8969\", \"SourceDestCheck\": \"true\", \"InterfaceType\": \"interface\", \"Tags\": []}, \"UpdateReplacePolicy\": \"Retain\", \"DeletionPolicy\": \"Retain\"}\",
  "AfterContext": "{\"Properties\":{\"Description\":\"Description\", \"PrivateIpAddress\":\"172.31.76.2\", \"PrivateIpAddresses\": [{\"PrivateIpAddress\": \"172.31.76.2\", \"Primary\": \"true\"}], \"SecondaryPrivateIpAddressCount\": \"0\", \"Ipv6PrefixCount\": \"0\", \"Ipv4PrefixCount\": \"0\", \"GroupSet\": [\"sg-05a45689b1059e82d\"], \"Ipv6Prefixes\": [], \"SubnetId\": \"subnet-455e8969\", \"SourceDestCheck\": \"true\", \"InterfaceType\": \"interface\", \"Tags\": [{\"Value\": \"Test\", \"Key\": \"Test\"}]}, \"UpdateReplacePolicy\": \"Retain\", \"DeletionPolicy\": \"Retain\"}"
}
  ],
  "ChangeSetName": "ExampleChangeSet",
  "ChangeSetId": "arn:aws:cloudformation:us-east-1:123456789012:changeSet/ExampleChangeSet/9f7b541b-126b-44f7-998e-932174557841",
  "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/ExampleStack/ab664180-f686-11ee-9e29-12cd92393671",

```

Jalankan set perubahan untuk CloudFormation tumpukan

Untuk membuat perubahan yang dijelaskan dalam set perubahan ke tumpukan Anda, jalankan set perubahan.

 Important

Setelah Anda menjalankan set perubahan, CloudFormation menghapus set perubahan tambahan yang terkait dengan tumpukan karena tidak lagi valid untuk tumpukan yang diperbarui. Jika pembaruan gagal, Anda perlu membuat set perubahan baru.

Kebijakan Tumpukan dan Menjalankan Set Perubahan

Jika Anda menjalankan set perubahan pada tumpukan yang memiliki kebijakan tumpukan yang berhubungan dengannya, CloudFormation memberlakukan kebijakan ketika memperbarui tumpukan. Anda tidak dapat menentukan kebijakan tumpukan sementara yang menimpa kebijakan yang ada saat Anda menjalankan set perubahan. Untuk memperbarui sumber daya yang dilindungi, Anda harus memperbarui kebijakan tumpukan atau menggunakan metode pembaruan langsung. Untuk informasi selengkapnya, lihat [Perbarui tumpukan secara langsung](#).

Execute a change set (console)

Untuk menjalankan set perubahan

1. Buka AWS CloudFormation konsol di <https://console.aws.amazon.com/cloudformation>.
2. Pada bilah navigasi di bagian atas layar, pilih Anda Wilayah AWS.
3. Pada halaman Stacks, pilih nama tumpukan yang ingin Anda perbarui.
4. Di panel navigasi, pilih Ubah set untuk melihat daftar set perubahan tumpukan.
5. Pilih nama set perubahan yang ingin Anda jalankan.
6. Pada halaman detail set perubahan, pilih Jalankan set perubahan.

CloudFormation segera mulai memperbarui tumpukan. CloudFormation Konsol mengarahkan Anda ke tab Acara, di mana Anda dapat memantau kemajuan pembaruan tumpukan. Untuk informasi selengkapnya, lihat [Pantau kemajuan tumpukan](#).

Execute a change set for nested stacks (console)

Untuk menjalankan set perubahan untuk tumpukan bersarang

1. Buka AWS CloudFormation konsol di <https://console.aws.amazon.com/cloudformation>.
2. Pada bilah navigasi di bagian atas layar, pilih Anda Wilayah AWS.

3. Pada halaman Stacks, pilih nama tumpukan yang ingin Anda perbarui. Anda harus memilih nama tumpukan yang berhubungan dengan set perubahan root.
4. Di panel navigasi, pilih Ubah set untuk melihat daftar set perubahan tumpukan.
5. Pilih nama set perubahan root yang ingin Anda jalankan.
6. Pada halaman detail set perubahan, pilih Jalankan set perubahan.

 Note

CloudFormation mengeksekusi perubahan yang dijelaskan dalam set perubahan root dan set perubahan bersarang, jika Diaktifkan untuk set perubahan untuk tumpukan bersarang dipilih selama proses. [Buat set perubahan untuk CloudFormation tumpukan](#)

CloudFormation segera mulai memperbarui tumpukan. CloudFormation Konsol mengarahkan Anda ke tab Acara, di mana Anda dapat memantau kemajuan pembaruan tumpukan. Untuk informasi selengkapnya, lihat [Pantau kemajuan tumpukan](#).

Untuk menjalankan set perubahan (AWS CLI)

- Jalankan [execute-change-set](#) perintah.

Tentukan ID set perubahan pada set perubahan yang ingin Anda jalankan, seperti yang ditunjukkan dalam contoh berikut:

```
aws cloudformation execute-change-set \  
  --change-set-name \  
    arn:aws:cloudformation:us-east-1:123456789012:changeSet/  
SampleChangeSet/1a2345b6-0000-00a0-a123-00abc0abc000
```

Perintah dalam contoh menjalankan set perubahan dengan `arn:aws:cloudformation:us-east-1:123456789012:changeSet/SampleChangeSet/1a2345b6-0000-00a0-a123-00abc0abc000` ID.

Setelah Anda menjalankan perintah, CloudFormation mulai memperbarui tumpukan. Untuk melihat kemajuan tumpukan, gunakan [perintah deskripsi-tumpukan](#).

Menghapus set perubahan untuk CloudFormation tumpukan

Menghapus set perubahan menghapusnya dari daftar set perubahan untuk tumpukan. Menghapus set perubahan mencegah Anda atau pengguna lain secara tidak sengaja menjalankan set perubahan yang seharusnya tidak diterapkan. Kecuali Anda menghapusnya, CloudFormation pertahankan semua set perubahan sampai Anda memperbarui tumpukan.

Delete a change set

Untuk menghapus set perubahan (konsol)

1. Buka AWS CloudFormation konsol di <https://console.aws.amazon.com/cloudformation>.
2. Pada bilah navigasi di bagian atas layar, pilih Anda Wilayah AWS.
3. Pada halaman Stacks, pilih nama tumpukan yang berisi set perubahan yang ingin Anda hapus.
4. Di panel navigasi, pilih Ubah set untuk melihat daftar set perubahan tumpukan.
5. Pilih nama set perubahan yang ingin Anda hapus.
6. Pada halaman detail set perubahan, pilih Hapus set perubahan.

CloudFormation segera mulai menghapus set perubahan dari daftar kumpulan perubahan tumpukan, dan Anda diarahkan ke halaman Stacks.

Delete a change set for nested stacks (console)

Untuk menghapus set perubahan untuk tumpukan bersarang

1. Buka AWS CloudFormation konsol di <https://console.aws.amazon.com/cloudformation>.
2. Pada bilah navigasi di bagian atas layar, pilih Anda Wilayah AWS.
3. Pada halaman Stacks, pilih nama tumpukan yang terkait dengan set perubahan root.
4. Di panel navigasi, pilih Ubah set untuk melihat daftar set perubahan tumpukan.
5. Pilih nama set perubahan yang ingin Anda hapus.
6. Pada halaman detail set perubahan, pilih Hapus. Dengan memilih Hapus set perubahan, Anda akan menghapus seluruh hierarki set perubahan bersarang.

Note

Operasi hapus untuk set perubahan untuk tumpukan nest adalah asinkron dan akan menunjukkan status `DELETE_PENDING`, diikuti oleh status `DELETE_IN_PROGRESS`. Setelah menyelesaikan operasi hapus set perubahan, set perubahan akan dihapus dari daftar. Tumpukan nest di status `REVIEW_IN_PROGRESS` juga akan dihapus jika dibuat selama pembuatan set perubahan.

CloudFormation segera mulai menghapus set perubahan dari daftar set perubahan tumpukan.

Note

Jika Anda memiliki tumpukan bersarang yang macet dalam operasi yang sedang berlangsung, lihat Memecahkan Masalah Kesalahan di [Kegagalan rollback tumpukan bersarang](#)

Untuk menghapus set perubahan (AWS CLI)

- Jalankan perintah [delete-change-set](#), menentukan ID set perubahan yang ingin Anda hapus, seperti yang ditunjukkan dalam contoh berikut:

```
aws cloudformation delete-change-set \
  --change-set-name \
    arn:aws:cloudformation:us-east-1:123456789012:changeSet/
SampleChangeSet/1a2345b6-0000-00a0-a123-00abc0abc000
```

Contoh set perubahan untuk CloudFormation tumpukan

Bagian ini memberikan contoh set perubahan yang CloudFormation akan dibuat untuk perubahan tumpukan umum. Contoh tersebut menunjukkan cara mengedit templat secara langsung; memodifikasi parameter input tunggal; merencanakan untuk pembuatan kembali sumber daya (penggantian), yang mencegah Anda kehilangan data yang tidak dicadangkan atau mengganggu aplikasi yang sedang berjalan di tumpukan Anda; dan menambah dan menghapus sumber daya.

Untuk menggambarkan bagaimana set perubahan bekerja, kita akan berjalan melalui perubahan yang dikirimkan dan membahas set perubahan yang dihasilkan. Karena setiap contoh dibuat berdasarkan dan mengasumsikan bahwa Anda memahami contoh sebelumnya, kami sarankan Anda membacanya secara berurutan. Untuk deskripsi setiap bidang dalam kumpulan perubahan, lihat [Mengubah](#) tipe data di Referensi AWS CloudFormation API.

Anda dapat menggunakan [konsol](#), AWS CLI, atau operasi CloudFormation [DescribeChangeSet](#) API untuk melihat detail set perubahan.

Kita menghasilkan setiap set perubahan berikut dari tumpukan dengan [templat sampel](#) berikut:

```
{
  "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" }
      }
    ]
  }
}
}
}

```

Mengedit templat secara langsung

Ketika Anda secara langsung memodifikasi sumber daya dalam templat tumpukan untuk menghasilkan set perubahan, CloudFormation mengklasifikasikan perubahan sebagai modifikasi langsung, dibandingkan dengan perubahan yang dimulai oleh nilai parameter yang diperbarui. Set perubahan berikut, yang menambahkan tanda baru ke instans `i-1abc23d4`, adalah contoh dari modifikasi langsung. Semua nilai input lainnya, seperti nilai dan kemampuan parameter, tidak berubah, jadi kita akan fokus pada struktur `Changes`.

```

{
  "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"
}

```

Di struktur `Changes`, hanya ada satu struktur `ResourceChange`. Struktur ini menjelaskan informasi seperti jenis sumber daya CloudFormation akan berubah, tindakan CloudFormation akan diambil, ID sumber daya, ruang lingkup perubahan, dan apakah perubahan memerlukan penggantian (di mana CloudFormation membuat sumber daya baru dan kemudian menghapus yang lama). Dalam contoh, set perubahan menunjukkan bahwa CloudFormation akan memodifikasi Tags atribut `i-1abc23d4` EC2 instance, dan tidak memerlukan instance untuk diganti.

Dalam `Details` struktur, CloudFormation label perubahan ini sebagai modifikasi langsung yang tidak akan pernah memerlukan instance untuk dibuat ulang (diganti). Anda dapat dengan percaya diri menjalankan perubahan ini, mengetahui bahwa itu CloudFormation tidak akan menggantikan instance.

CloudFormation Menunjukkan perubahan ini sebagai `Static` evaluasi. Evaluasi statis berarti CloudFormation dapat menentukan nilai tag sebelum mengeksekusi set perubahan. Dalam beberapa kasus, CloudFormation dapat menentukan nilai hanya setelah Anda menjalankan set perubahan.

CloudFormation melabeli perubahan tersebut sebagai *Dynamic evaluasi*. Misalnya, jika Anda mereferensikan sumber daya yang diperbarui yang diganti secara kondisional, tidak CloudFormation dapat menentukan apakah referensi ke sumber daya yang diperbarui akan berubah.

Memodifikasi nilai parameter input

Saat Anda memodifikasi nilai parameter input, CloudFormation menghasilkan dua perubahan untuk setiap sumber daya yang menggunakan nilai parameter yang diperbarui. Dalam contoh ini, kita ingin menyoroti seperti apa perubahan tersebut dan informasi mana yang harus Anda fokuskan. Contoh berikut dihasilkan dengan hanya mengubah nilai parameter input *Purpose*.

PurposeParameter menentukan nilai kunci tag untuk EC2 contoh. Dalam contoh, nilai parameter diubah dari *testing* ke *production*. Nilai baru ditunjukkan dalam struktur *Parameters*.

```
{
  "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"
}

```

ChangesStruktur berfungsi mirip dengan cara yang dilakukannya dalam [Mengedit templat secara langsung](#) contoh. Hanya ada satu ResourceChange struktur; itu menggambarkan perubahan pada Tags atribut i-1abc23d4 EC2 instance.

Namun, dalam struktur Details, set perubahan menunjukkan dua perubahan untuk atribut Tags, meskipun hanya satu nilai parameter yang diubah. Sumber daya yang mereferensikan nilai parameter yang diubah (menggunakan fungsi intrinsik Ref) selalu menghasilkan dua perubahan: satu dengan evaluasi Dynamic dan lainnya dengan evaluasi Static. Anda dapat melihat tipe perubahan ini dengan melihat bidang berikut:

- Untuk perubahan evaluasi `Static`, lihat bidang `ChangeSource`. Dalam contoh ini, bidang `ChangeSource` sama dengan `ParameterReference`, yang berarti bahwa perubahan ini adalah hasil dari nilai referensi parameter yang diperbarui. Set perubahan harus berisi perubahan evaluasi `Dynamic` yang serupa.
- Anda dapat menemukan perubahan evaluasi `Dynamic` yang cocok dengan membandingkan struktur `Target` untuk kedua perubahan, yang akan berisi informasi yang sama. Dalam contoh ini, struktur `Target` untuk kedua perubahan berisi nilai yang sama untuk bidang `Attribute` dan `RequireRecreation`.

Untuk tipe perubahan ini, fokuskan pada evaluasi statis, yang memberi Anda informasi paling rinci tentang perubahan. Dalam contoh ini, evaluasi statis menunjukkan bahwa perubahan adalah hasil perubahan pada nilai referensi parameter (`ParameterReference`). Parameter yang tepat yang diubah ditunjukkan oleh bidang `CauseEntity` (parameter `Purpose`).

Menentukan nilai bidang penggantian

`ReplacementBidang` dalam `ResourceChange` struktur menunjukkan apakah CloudFormation akan membuat ulang sumber daya. Merencanakan pembuatan ulang sumber daya (penggantian) mencegah Anda kehilangan data yang tidak dicadangkan atau mengganggu aplikasi yang berjalan di tumpukan Anda.

Nilai dalam bidang `Replacement` tergantung pada apakah perubahan memerlukan penggantian, ditunjukkan dengan bidang `RequiresRecreation` dalam struktur `Target` perubahan. Sebagai contoh, jika bidang `RequiresRecreation` adalah `Never`, bidang `Replacement` adalah `False`. Namun, jika ada beberapa perubahan pada satu sumber daya dan setiap perubahan memiliki nilai yang berbeda untuk `RequiresRecreation` bidang tersebut, CloudFormation perbarui sumber daya menggunakan perilaku yang paling mengganggu. Dengan kata lain, jika hanya satu dari banyak perubahan memerlukan penggantian, CloudFormation harus mengganti sumber daya dan, oleh karena itu, mengatur bidang `Replacement` ke `True`.

Set perubahan berikut dihasilkan dengan mengubah nilai untuk setiap parameter (`Purpose`, `InstanceType`, dan `KeyPairName`), yang semuanya digunakan oleh EC2 instance. Dengan perubahan ini, CloudFormation akan diperlukan untuk mengganti instance karena `Replacement` bidang sama dengan `True`.

```
{
  "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
  MyStack/1a2345b6-0000-00a0-a123-00abc0abc000",
```

```
"Status": "CREATE_COMPLETE",
"ChangeSetName": "SampleChangeSet-multiple",
"Parameters": [
  {
    "ParameterValue": "production",
    "ParameterKey": "Purpose"
  },
  {
    "ParameterValue": "MyNewKeyName",
    "ParameterKey": "KeyPairName"
  },
  {
    "ParameterValue": "t2.small",
    "ParameterKey": "InstanceType"
  }
],
"Changes": [
  {
    "ResourceChange": {
      "ResourceType": "AWS::EC2::Instance",
      "PhysicalResourceId": "i-7bef86f8",
      "Details": [
        {
          "ChangeSource": "DirectModification",
          "Evaluation": "Dynamic",
          "Target": {
            "Attribute": "Properties",
            "Name": "KeyName",
            "RequiresRecreation": "Always"
          }
        },
        {
          "ChangeSource": "DirectModification",
          "Evaluation": "Dynamic",
          "Target": {
            "Attribute": "Properties",
            "Name": "InstanceType",
            "RequiresRecreation": "Conditionally"
          }
        }
      ],
      {
        "ChangeSource": "DirectModification",
        "Evaluation": "Dynamic",
        "Target": {
```

```

        "Attribute": "Tags",
        "RequiresRecreation": "Never"
    }
},
{
    "CausingEntity": "KeyPairName",
    "ChangeSource": "ParameterReference",
    "Evaluation": "Static",
    "Target": {
        "Attribute": "Properties",
        "Name": "KeyName",
        "RequiresRecreation": "Always"
    }
},
{
    "CausingEntity": "InstanceType",
    "ChangeSource": "ParameterReference",
    "Evaluation": "Static",
    "Target": {
        "Attribute": "Properties",
        "Name": "InstanceType",
        "RequiresRecreation": "Conditionally"
    }
},
{
    "CausingEntity": "Purpose",
    "ChangeSource": "ParameterReference",
    "Evaluation": "Static",
    "Target": {
        "Attribute": "Tags",
        "RequiresRecreation": "Never"
    }
}
],
"Action": "Modify",
"Scope": [
    "Tags",
    "Properties"
],
"LogicalResourceId": "MyEC2Instance",
"Replacement": "True"
},
"Type": "Resource"
}

```

```
],
"CreationTime": "2020-11-18T00:39:35.974Z",
"Capabilities": [],
"StackName": "MyStack",
"NotificationARNs": [],
"ChangeSetId": "arn:aws:cloudformation:us-east-1:123456789012:changeSet/
SampleChangeSet-multiple/1a2345b6-0000-00a0-a123-00abc0abc000"
}
```

Identifikasi perubahan yang memerlukan sumber daya untuk diganti dengan melihat setiap perubahan (evaluasi statis di struktur `Details`). Dalam contoh ini, setiap perubahan memiliki nilai yang berbeda untuk `RequireRecreation` bidang tersebut, tetapi perubahan pada `KeyName` properti memiliki perilaku pembaruan yang paling mengganggu, selalu membutuhkan rekreasi. CloudFormation akan menggantikan instance karena nama kunci diubah.

Jika nama kunci tidak berubah, perubahan ke properti `InstanceType` akan memiliki perilaku pembaruan yang paling mengganggu (`Conditionally`), sehingga bidang `Replacement` akan `Conditionally`. Untuk menemukan kondisi yang CloudFormation menggantikan instance, lihat perilaku pembaruan untuk `InstanceType` properti jenis [AWS::EC2::Instances](#) sumber daya.

Menambahkan dan menghapus sumber daya

Contoh berikut dihasilkan dengan mengirimkan template yang dimodifikasi yang menghapus EC2 instance dan menambahkan grup Auto Scaling dan konfigurasi peluncuran.

```
{
  "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"
}

```

Di struktur `Changes`, ada tiga struktur `ResourceChange`, satu untuk setiap sumber daya. Untuk setiap sumber daya, `Action` bidang menunjukkan apakah CloudFormation menambah atau menghapus sumber daya. Bidang `Scope` dan `Details` kosong karena hanya berlaku untuk sumber daya yang dimodifikasi.

Untuk sumber daya baru, tidak CloudFormation dapat menentukan nilai beberapa bidang hingga Anda menjalankan set perubahan. Misalnya, CloudFormation tidak menyediakan fisik IDs grup Auto Scaling dan konfigurasi peluncuran karena belum ada. CloudFormation membuat sumber daya baru saat Anda menjalankan set perubahan.

Melihat perubahan tingkat properti

Contoh berikut menunjukkan perubahan tingkat Tag properti pada properti instance Amazon EC2 . Tag `Value` dan `Key` akan berubah menjadi `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": "KeyName",
      "ParameterValue": "BatchTest"
    },
    {
      "ParameterKey": "Purpose",
      "ParameterValue": "testing"
    },
    {
      "ParameterKey": "InstanceType",
      "ParameterValue": "t2.micro"
    }
  ],
  "CreationTime": "2024-04-09T21:29:10.759000+00:00",
  "ExecutionStatus": "AVAILABLE",
  "Status": "CREATE_COMPLETE",
  "StatusReason": null,
  "NotificationARNs": [],
  "RollbackConfiguration": {
```



```

: ...skipping...
{
  "Changes": [
    {
      "Type": "Resource",
      "ResourceChange": {
        "Action": "Modify",
        "LogicalResourceId": "MyEC2Instance",
        "PhysicalResourceId": "i-0cc7856a36315e62b",
        "ResourceType": "AWS::EC2::Instance",
        "Replacement": "False",
        "Scope": [
          "Tags"
        ],
        "Details": [
          {
            "Target": {
              "Attribute": "Tags",
              "RequiresRecreation": "Never",
              "Path": "/Properties/Tags/0/Value",
              "BeforeValue": "testing",
              "AfterValue": "Test",
              "AttributeChangeType": "Modify"
            },
            "Evaluation": "Static",
            "ChangeSource": "DirectModification"
          },
          {
            "Target": {
              "Attribute": "Tags",
              "RequiresRecreation": "Never",
              "Path": "/Properties/Tags/0/Key",
              "BeforeValue": "Purpose",
              "AfterValue": "Test",
              "AttributeChangeType": "Modify"
            },
            "Evaluation": "Static",
            "ChangeSource": "DirectModification"
          }
        ],
        "BeforeContext": "{\"Properties\":{\"KeyName\":\"BatchTest\",\"ImageId\":\"ami-8fcee4e5\",\"InstanceType\":\"t2.micro\",\"Tags\":[{\"Value\":\"testing\",\"Key\":\"Purpose\"]}}}"
      }
    }
  ]
}

```

```

    "AfterContext": "{ \"Properties\": { \"KeyName\": \"BatchTest\", \"ImageId\": \"ami-8fcee4e5\", \"InstanceType\": \"t2.micro\", \"Tags\": [ { \"Value\": \"Test\", \"Key\": \"Test\" } ] } }"
  }
}
]

```

DetailsStruktur menunjukkan nilai untuk Key dan Value sebelum set perubahan dijalankan, dan apa yang akan terjadi setelah set perubahan dijalankan.

Set perubahan untuk tumpukan yang di-nest

Dengan set perubahan untuk tumpukan yang di-nest Anda dapat melihat pratinjau perubahan ke sumber daya aplikasi dan infrastruktur Anda di seluruh hierarki tumpukan yang di-nest dan melanjutkan dengan pembaruan ketika Anda telah mengonfirmasi bahwa semua perubahan seperti yang dimaksudkan.

Lihat bagian berikut untuk detail selengkapnya tentang set perubahan untuk tumpukan yang di-nest:

Topik

- [Gambaran umum set perubahan dan tumpukan yang di-nest](#)
- [Menggunakan set perubahan untuk tumpukan yang di-nest \(konsol\)](#)
- [Menggunakan set perubahan untuk tumpukan yang di-nest \(AWS CLI\)](#)

Gambaran umum set perubahan dan tumpukan yang di-nest

Set perubahan untuk tumpukan yang di-nest menggabungkan fitur berikut bersama-sama untuk memperluas cakupan pratinjau perubahan ke seluruh hierarki tumpukan:

- Set perubahan adalah CloudFormation kemampuan yang menawarkan pratinjau tentang bagaimana perubahan yang diusulkan ke tumpukan akan berdampak pada sumber daya yang ada atau yang baru dibuat. Setelah membuat set perubahan, CloudFormation berikan daftar perubahan yang diusulkan dengan membandingkan tumpukan Anda dengan perubahan pada sumber daya yang Anda kirimkan. Untuk informasi selengkapnya tentang set perubahan, lihat [Perbarui CloudFormation tumpukan menggunakan set perubahan](#).
- Tumpukan bersarang adalah tumpukan yang dibuat sebagai bagian dari tumpukan lain. Misalnya, Anda mungkin memiliki sumber daya terkait jaringan dan keamanan di satu tumpukan yang di-nest dan sumber daya aplikasi di tumpukan lain. Mempartisi model aplikasi dengan cara ini akan

membantu pemeliharaan dan penggunaan kembali kode. Untuk informasi lebih lanjut tentang tumpukan bersarang, lihat. [Pisahkan template menjadi potongan-potongan yang dapat digunakan kembali menggunakan tumpukan bersarang](#)

Menggunakan set perubahan untuk tumpukan yang di-nest (konsol)

- Buat set perubahan - Membuat set perubahan dengan mengirimkan perubahan dari setiap tingkat hierarki tumpukan. Anda dapat mengirimkan templat tumpukan yang diubah atau nilai parameter yang diubah dan CloudFormation membandingkan tumpukan yang di-nest Anda dengan perubahan yang Anda kirimkan untuk menghasilkan set perubahan. Set perubahan untuk tumpukan yang di-nest diaktifkan secara default di konsol CloudFormation . Untuk informasi selengkapnya, lihat [Buat set perubahan untuk CloudFormation tumpukan](#).

▼ 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

Set perubahan root adalah set perubahan terkait dengan tumpukan asal dari seluruh hierarki set perubahan dibuat. Anda harus menjalankan atau menghapus set perubahan untuk tumpukan yang di-nest dari set perubahan root. Untuk informasi selengkapnya, lihat [Melakukan operasi tumpukan pada tumpukan bersarang](#).

- Lihat set perubahan — Visualisasikan perubahan pada sumber daya di dalam tumpukan bersarang sebelum menjalankannya. Anda dapat melihat perubahan yang diajukan di bagian Perubahan dari set perubahan dengan menavigasi melalui tumpukan saat ini dan set perubahan yang di-nest-nya. Untuk informasi selengkapnya, lihat [Melihat set perubahan untuk CloudFormation tumpukan](#).
- Jalankan set perubahan - Jalankan perubahan yang dijelaskan dalam set perubahan yang berkaitan dengan tumpukan saat ini dan keturunannya. Operasi eksekusi harus dilakukan dari set perubahan root. Untuk informasi selengkapnya, lihat [Jalankan set perubahan untuk CloudFormation tumpukan](#).
- Hapus set perubahan - Menghapus set perubahan dari tumpukan saat ini. Menghapus set perubahan membantu mencegah Anda atau pengguna lain secara tidak sengaja memulai set

perubahan yang seharusnya tidak diterapkan. Operasi hapus harus dijalankan dari set perubahan root. Untuk informasi selengkapnya, lihat [Menghapus set perubahan untuk CloudFormation tumpukan](#).

Menggunakan set perubahan untuk tumpukan yang di-nest (AWS CLI)

create-change-set

- [create-change-set](#)— Ubah set untuk tumpukan bersarang tidak diaktifkan secara default untuk AWS CLI. Untuk membuat set perubahan untuk seluruh hierarki tumpukan, tentukan `--include-nested-stacks` opsi. Untuk informasi selengkapnya, lihat [Buat set perubahan untuk CloudFormation tumpukan](#).

AWS CLI Contoh berikut membuat set perubahan untuk tumpukan root yang ditentukan.

```
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
```

Berikut ini adalah output contoh.

```
{
  "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](#)— Mengembalikan daftar perubahan yang CloudFormation akan dilakukan jika Anda menjalankan set perubahan. Jika set perubahan yang ditentukan berisi set perubahan turunan yang dimiliki tumpukan yang di-nest, maka `ChangeSetId` akan menampilkan informasi tentang set perubahan tersebut. Untuk informasi selengkapnya, lihat [Melihat set perubahan untuk CloudFormation tumpukan](#).

AWS CLI Contoh berikut menjelaskan set perubahan untuk tumpukan root yang ditentukan.

```
aws cloudformation describe-change-set \
  --change-set-name my-root-stack-change-set \
  --stack-name my-root-stack
```

Berikut ini adalah output contoh.

```
{
  "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
}

```

AWS CLI Contoh berikut menjelaskan set perubahan untuk tumpukan bersarang yang ditentukan.

```

aws cloudformation describe-change-set \
  --change-set-name my-nested-stack-change-set \
  --stack-name my-nested-stack

```

Berikut ini adalah output contoh.

```

{
  "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](#)— Membuat atau memperbarui tumpukan menggunakan informasi masukan yang disediakan saat set perubahan yang ditentukan dibuat. Untuk membuat set perubahan untuk seluruh hierarki tumpukan, Anda harus menentukan `--include-nested-stacks` opsi selama `create-change-set` operasi. Untuk informasi selengkapnya, lihat [Jalankan set perubahan untuk CloudFormation tumpukan](#).

Note

`execute-change-set` harus dijalankan dari set perubahan root dan akan menerapkan set perubahan pada seluruh hierarki tumpukan.

AWS CLI Contoh berikut mengeksekusi set perubahan untuk tumpukan root yang ditentukan.

```
aws cloudformation execute-change-set \  
  --stack-name my-root-stack \  
  --change-set-name my-root-stack-change-set
```

delete-change-set

- [delete-change-set](#)— Menghapus set perubahan yang ditentukan. Menghapus set perubahan memastikan bahwa tidak ada yang menggunakan set perubahan yang salah. Menghapus set perubahan tidak sinkron untuk set perubahan yang dibuat dengan opsi. `--include-nested-stacks` Untuk informasi selengkapnya, lihat [Menghapus set perubahan untuk CloudFormation tumpukan](#).

Note

`delete-change-set` harus dijalankan dari set perubahan root dan akan menghapus seluruh hierarki set perubahan. Tumpukan bersarang dalam `REVIEW_IN_PROGRESS` status juga akan dihapus jika dibuat selama operasi. `create-change-set`

AWS CLI Contoh berikut menghapus set perubahan untuk tumpukan root yang ditentukan.

```
aws cloudformation delete-change-set \  
  --stack-name my-root-stack \  
  --change-set-name my-root-stack-change-set
```



```
--change-set-name my-root-stack-change-set
```

Perbarui tumpukan secara langsung

Bila ingin men-deploy pembaruan ke tumpukan Anda dengan cepat, lakukan pembaruan langsung. Dengan pembaruan langsung, Anda mengirimkan templat atau parameter input yang menentukan pembaruan ke sumber daya di tumpukan, dan CloudFormation segera menerapkannya. Jika Anda ingin menggunakan template untuk membuat pembaruan, Anda dapat memodifikasi template saat ini dan menyimpannya secara lokal atau di bucket Amazon S3.

Untuk atribut sumber daya yang tidak mendukung pembaruan, Anda harus menjaga nilai saat ini. Untuk melihat pratinjau perubahan yang CloudFormation akan dilakukan pada tumpukan Anda sebelum Anda memperbaruinya, gunakan set perubahan. Untuk informasi selengkapnya, lihat [Perbarui CloudFormation tumpukan menggunakan set perubahan](#).

Saat memperbarui tumpukan, CloudFormation mungkin mengganggu sumber daya atau mengganti sumber daya yang diperbarui, tergantung pada properti yang Anda perbarui. Untuk informasi selengkapnya tentang perilaku pembaruan sumber daya, lihat [Memahami perilaku pembaruan sumber daya tumpukan](#).

Untuk memperbarui tumpukan (konsol)

1. Masuk ke AWS Management Console dan buka AWS CloudFormation konsol di <https://console.aws.amazon.com/cloudformation>.
2. Pada bilah navigasi di bagian atas layar, pilih Anda Wilayah AWS.
3. Pada halaman Stacks, pilih tumpukan berjalan yang ingin Anda perbarui.
4. Di panel detail tumpukan, pilih Perbarui.
5. Jika Anda belum memodifikasi template tumpukan, pilih Gunakan template yang ada, lalu pilih Berikutnya.

Jika Anda telah memodifikasi templat, pilih Ganti templat yang ada dan tentukan lokasi templat yang diperbarui di Tentukan templat bagian:

- Untuk templat yang disimpan secara lokal di komputer Anda, pilih Unggah file templat. Pilih dile untuk menavigasi ke file dan memilihnya, lalu pilih Selanjutnya.

 Note

Jika Anda mengunggah file template lokal, CloudFormation unggah ke bucket Amazon Simple Storage Service (Amazon S3) di bucket Anda. Akun AWS Jika Anda belum memiliki bucket S3 yang dibuat oleh CloudFormation, itu akan membuat bucket unik untuk setiap Wilayah tempat Anda mengunggah file template. Jika Anda sudah memiliki bucket S3 yang dibuat oleh CloudFormation Anda Akun AWS, CloudFormation tambahkan template ke bucket itu.

Pertimbangan yang perlu diingat tentang ember S3 yang dibuat oleh CloudFormation

- Bucket dapat diakses oleh siapa saja yang memiliki izin Amazon S3 di Anda. Akun AWS
- CloudFormation membuat bucket dengan enkripsi sisi server diaktifkan secara default, sehingga mengenkripsi semua objek yang disimpan dalam bucket.

Anda dapat langsung mengelola opsi enkripsi untuk bucket yang CloudFormation telah dibuat; misalnya, menggunakan konsol Amazon S3 <https://console.aws.amazon.com/s3/>di, atau. AWS CLI Untuk informasi selengkapnya, lihat [Menyetel perilaku enkripsi sisi server default untuk bucket Amazon S3 di](#) Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

- Anda dapat menggunakan bucket Anda sendiri dan mengelola izinnya dengan mengunggah templat secara manual ke Amazon S3. Saat Anda membuat atau memperbarui tumpukan, tentukan URL Amazon S3 dari file templat.

- Untuk template yang disimpan dalam bucket Amazon S3, pilih URL Amazon S3. Masukkan atau tempel URL untuk templat, lalu pilih Selanjutnya.

Jika Anda memiliki template dalam bucket berkemampuan versi, Anda dapat menentukan versi template tertentu dengan menambahkan `?versionId=version-id` ke URL. Untuk informasi selengkapnya, lihat [Bekerja dengan objek dalam bucket berkemampuan versi](#) di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

Jika ada masalah sintaks yang terdeteksi, konsol menyediakan pesan kesalahan yang membantu Anda memperbaiki template.

6. Jika templat Anda berisi parameter, di halaman Tentukan detail tumpukan Anda dapat memasukkan atau mengubah nilai parameter, lalu memilih Selanjutnya.

CloudFormation mengisi setiap parameter dengan nilai yang saat ini ditetapkan dalam tumpukan dengan pengecualian parameter yang dideklarasikan dengan NoEcho atribut; Namun, Anda masih dapat menggunakan nilai saat ini dengan memeriksa Gunakan nilai yang ada.

Untuk informasi selengkapnya tentang penggunaan NoEcho untuk menutupi informasi sensitif, selain menggunakan parameter dinamis untuk mengelola rahasia, lihat praktik [Jangan menanamkan kredensial dalam templat Anda](#) terbaik.

7. Pada halaman Configure stack options, Anda dapat memperbarui tag dan izin yang diterapkan ke tumpukan, dan memodifikasi opsi lanjutan seperti kebijakan tumpukan, konfigurasi rollback, atau memperbarui topik notifikasi Amazon SNS. Untuk informasi selengkapnya tentang opsi ini, lihat [Konfigurasi opsi tumpukan](#).
8. Jika template Anda berisi sumber daya IAM, untuk Kemampuan, pilih Saya mengakui bahwa templat ini dapat membuat sumber daya IAM untuk menentukan bahwa Anda ingin menggunakan sumber daya IAM dalam templat. Untuk informasi selengkapnya, lihat [Mengakui sumber daya IAM dalam templat CloudFormation](#).
9. Pilih Next untuk melanjutkan.
10. Tinjau informasi tumpukan dan perubahan yang Anda kirimkan.

Periksa bahwa Anda telah mengirimkan informasi yang benar, seperti nilai parameter atau URL templat yang benar.

Di bagian Ubah set pratinjau, periksa apakah CloudFormation akan membuat semua perubahan yang Anda harapkan. Misalnya, Anda dapat memeriksa apakah CloudFormation menambahkan, menghapus, dan memodifikasi sumber daya yang ingin Anda tambahkan, hapus, atau modifikasi. CloudFormation menghasilkan pratinjau ini dengan membuat set perubahan untuk tumpukan. Untuk informasi selengkapnya, lihat [Perbarui CloudFormation tumpukan menggunakan set perubahan](#).

11. Ketika Anda puas dengan perubahan Anda, pilih Perbarui tumpukan.

 Note

Pada titik ini, Anda juga memiliki opsi untuk melihat set perubahan untuk meninjau pembaruan yang diajukan dengan lebih teliti. Untuk melakukannya, pilih Lihat perubahan set bukan Update stack. CloudFormation menampilkan set perubahan yang dihasilkan

berdasarkan pembaruan Anda. Jika Anda siap untuk melakukan pembaruan tumpukan, pilih Jalankan.

CloudFormation menampilkan halaman detail tumpukan untuk tumpukan Anda, dengan panel Acara dipilih. Tumpukan Anda sekarang memiliki status `UPDATE_IN_PROGRESS`. Setelah CloudFormation berhasil selesai memperbarui tumpukan, ia menetapkan status tumpukan ke `UPDATE_COMPLETE`.

Jika pembaruan tumpukan gagal, CloudFormation; secara otomatis memutar kembali perubahan, dan menetapkan status tumpukan ke `UPDATE_ROLLBACK_COMPLETE`.

 Note

Anda dapat membatalkan pembaruan saat berada di `UPDATE_IN_PROGRESS` negara bagian. Untuk informasi selengkapnya, lihat [Batalan pembaruan tumpukan](#).

Untuk memperbarui tumpukan menggunakan baris perintah

Anda dapat menggunakan salah satu perintah berikut:

- [update-stack](#) (AWS CLI)
- [Update-CFNStack](#) (AWS Tools for Windows PowerShell)

Untuk contoh menggunakan baris perintah untuk memperbarui tumpukan, lihat [Contoh perintah operasi CloudFormation tumpukan untuk AWS CLI dan PowerShell](#).

Batalan pembaruan tumpukan

Setelah pembaruan tumpukan telah dimulai, Anda dapat membatalkan pembaruan tumpukan jika tumpukan masih dalam status `UPDATE_IN_PROGRESS`. Setelah pembaruan selesai, Anda tidak dapat membatalkannya. Akan tetapi, Anda dapat memperbarui tumpukan lagi dengan pengaturan sebelumnya.

Jika Anda membatalkan pembaruan tumpukan, tumpukan dikembalikan ke konfigurasi tumpukan yang ada sebelum memulai pembaruan tumpukan.

Topik

- [Untuk membatalkan pembaruan tumpukan \(konsol\)](#)
- [Untuk membatalkan pembaruan tumpukan \(AWS CLI\)](#)

Untuk membatalkan pembaruan tumpukan (konsol)

1. Masuk ke AWS Management Console dan buka AWS CloudFormation konsol di <https://console.aws.amazon.com/cloudformation>.
2. Pada bilah navigasi di bagian atas layar, pilih Wilayah AWS tempat tumpukan berada.
3. Pada halaman Stacks, pilih tumpukan yang sedang diperbarui. Statusnya harus UPDATE_IN_PROGRESS.
4. Pilih Tindakan tumpukan kemudian Batalkan tumpukan pembaruan.
5. Untuk melanjutkan membatalkan pembaruan, pilih Batalkan pembaruan. Jika tidak, pilih Batalkan untuk melanjutkan pembaruan.

Tumpukan diproses ke UPDATE_ROLLBACK_IN_PROGRESS status. Setelah pembatalan pembaruan selesai, tumpukan diatur ke UPDATE_ROLLBACK_COMPLETE.

Untuk membatalkan pembaruan tumpukan (AWS CLI)

Gunakan perintah [cancel-update-stack](#) untuk membatalkan pembaruan. Untuk informasi selengkapnya, lihat [Batalkan pembaruan tumpukan](#).

Hapus tumpukan dari CloudFormation konsol

Jika Anda tidak lagi membutuhkan sumber daya dalam tumpukan, Anda dapat menghapus seluruh tumpukan.

Saat menghapus tumpukan, CloudFormation menghapus semua sumber daya di tumpukan tersebut kecuali Anda menggunakan kebijakan penghapusan untuk mempertahankan sumber daya tertentu. Untuk informasi lebih lanjut, lihat [Atribut DeletionPolicy](#).

Untuk menghapus tumpukan (konsol)

1. Masuk ke AWS Management Console dan buka AWS CloudFormation konsol di <https://console.aws.amazon.com/cloudformation>.

2. Pada bilah navigasi di bagian atas layar, pilih Wilayah AWS tempat tumpukan berada.
3. Pada halaman Stacks, pilih tumpukan yang ingin Anda hapus. Tumpukan pasti sedang berjalan.
4. Pilih Hapus.
5. Ketika diminta konfirmasi, pilih Hapus.

 Note

Operasi penghapusan tumpukan tidak dapat dihentikan setelah penghapusan tumpukan dimulai. Tumpukan diproses ke DELETE_IN_PROGRESS status.

Setelah penghapusan tumpukan selesai, tumpukan akan berada di DELETE_COMPLETE status. Tumpukan dalam DELETE_COMPLETE status tidak ditampilkan di CloudFormation konsol secara default. Untuk menampilkan tumpukan yang dihapus, Anda harus mengubah tumpukan tampilan filter seperti yang dijelaskan di [Lihat tumpukan yang dihapus dari konsol CloudFormation](#).

Untuk memaksa menghapus tumpukan (konsol)

Penghapusan tumpukan mungkin gagal karena sumber daya di tumpukan gagal dihapus. Misalnya, CloudFormation akan gagal penghapusan sumber daya yang tumpukan lain juga bergantung pada. Sumber daya apa pun yang belum dihapus akan tetap ada hingga Anda berhasil menghapus tumpukan. Jika penghapusan gagal dan mengembalikan DELETE_FAILED status, Anda dapat memilih untuk mencoba lagi menggunakan salah satu dari dua metode.

1. Pada halaman Stacks di CloudFormation konsol, pilih tumpukan yang ingin Anda hapus paksa.
2. Di panel detail tumpukan, pilih Coba lagi hapus.
3. Pilih di antara opsi berikut:
 - Hapus tumpukan ini tetapi pertahankan sumber daya: Opsi ini memungkinkan Anda memilih sumber daya tertentu yang awalnya gagal dihapus, tetapi Anda ingin mempertahankan selama penghapusan tumpukan paksa.
 - Hapus paksa seluruh tumpukan ini: Opsi ini mempertahankan semua sumber daya yang gagal dihapus.
4. Pilih Hapus untuk memulai proses penghapusan paksa dengan pilihan Anda.

Untuk meninjau sumber daya yang dipertahankan (konsol)

Setelah menghapus tumpukan, Anda dapat melihat sumber daya yang disimpan di konsol.

1. Dalam daftar tumpukan, pilih status Filter dan pilih Dihapus.
2. Pilih tumpukan yang dihapus.
3. Pilih tab Sumber Daya.
4. Semua sumber daya yang dipertahankan menunjukkan DELETE_SKIPPED Status.
5. Pilih sumber daya yang disimpan yang ingin Anda tinjau.

Untuk menghapus tumpukan menggunakan baris perintah

Anda dapat menggunakan salah satu perintah berikut:

- [delete-stack](#) (AWS CLI)
- [Remove-CFNStack](#) (AWS Tools for Windows PowerShell)

Untuk contoh menggunakan baris perintah untuk menghapus tumpukan, lihat [Contoh perintah operasi CloudFormation tumpukan untuk AWS CLI dan PowerShell](#).

Sumber daya terkait

Untuk membantu memecahkan masalah kesalahan penghapusan tumpukan, lihat topik pemecahan masalah. [Gagal menghapus tumpukan](#)

Untuk informasi tentang melindungi tumpukan agar tidak terhapus secara tidak sengaja, lihat [Lindungi CloudFormation tumpukan agar tidak dihapus](#).

Lihat tumpukan yang dihapus dari konsol CloudFormation

Secara default, CloudFormation konsol tidak menampilkan tumpukan dengan status.

DELETE_COMPLETE Untuk menampilkan informasi tentang tumpukan yang dihapus, Anda harus mengubah tampilan tumpukan.

Untuk melihat tumpukan yang dihapus

1. Masuk ke AWS Management Console dan buka AWS CloudFormation konsol di <https://console.aws.amazon.com/cloudformation>.

2. Pada bilah navigasi di bagian atas layar, pilih Wilayah AWS tempat tumpukan yang dihapus berada.
3. Pada halaman Stacks, pilih Dihapus dari drop-down status Filter.

CloudFormation daftar semua tumpukan Anda dengan status. DELETE_COMPLETE

Lihat juga

- [Hapus tumpukan dari CloudFormation konsol](#)
- [Lihat informasi tumpukan dari CloudFormation konsol](#)

Pantau kemajuan tumpukan

Bagian ini menjelaskan cara memantau penyebaran tumpukan yang sedang berlangsung. CloudFormation menyediakan daftar peristiwa penyebaran yang terperinci dan kronologis, yang menunjukkan kemajuan dan masalah apa pun yang dihadapi selama penerapan.

Topik

- [Lihat acara CloudFormation tumpukan](#)
- [Melihat garis waktu penerapan CloudFormation tumpukan](#)
- [Memahami peristiwa pembuatan CloudFormation tumpukan](#)
- [Pantau kemajuan pembaruan tumpukan](#)
- [Terus mengembalikan pembaruan](#)
- [Tentukan penyebab kegagalan tumpukan](#)
- [Pilih cara menangani kegagalan saat menyediakan sumber daya](#)

Lihat acara CloudFormation tumpukan

Peristiwa tumpukan membantu Anda memahami kapan sumber daya sedang dibuat, diperbarui, atau dihapus, dan apakah operasi tumpukan berjalan seperti yang diharapkan. Anda dapat melihat peristiwa tumpukan untuk memantau kemajuan dan status AWS CloudFormation tumpukan Anda.

Topik

- [Lihat acara tumpukan \(konsol\)](#)
- [Lihat acara tumpukan \(AWS CLI\)](#)

- [Kode status tumpukan](#)

Lihat acara tumpukan (konsol)

Untuk melihat peristiwa tumpukan

1. Buka AWS CloudFormation konsol di <https://console.aws.amazon.com/cloudformation>.
2. Pada bilah navigasi di bagian atas layar, pilih tumpukan yang Wilayah AWS Anda buat.
3. Pada halaman Stacks CloudFormation konsol, pilih nama tumpukan. CloudFormation menampilkan detail tumpukan untuk tumpukan yang dipilih.
4. Pilih tab Acara untuk melihat peristiwa tumpukan yang CloudFormation telah dihasilkan untuk tumpukan Anda.

CloudFormation secara otomatis menyegarkan peristiwa tumpukan setiap menit. Selain itu, CloudFormation menampilkan lencana Acara baru yang tersedia saat peristiwa tumpukan baru terjadi. Pilih ikon penyegaran untuk memuat peristiwa ini ke dalam daftar. Dengan melihat acara pembuatan tumpukan, Anda dapat memahami urutan acara yang mengarah ke pembuatan tumpukan Anda (atau kegagalan, jika Anda mengawakutu tumpukan Anda).

Saat tumpukan Anda sedang dibuat, itu terdaftar di halaman Stacks dengan status. `CREATE_IN_PROGRESS` Setelah tumpukan Anda berhasil dibuat, statusnya berubah menjadi `CREATE_COMPLETE`.

Untuk informasi selengkapnya, silakan lihat [Memahami peristiwa pembuatan CloudFormation tumpukan](#) dan [Pantau kemajuan pembaruan tumpukan](#).

Lihat acara tumpukan (AWS CLI)

Atau, Anda dapat menggunakan [describe-stack-events](#) perintah saat tumpukan sedang dibuat untuk melihat peristiwa seperti yang dilaporkan.

`describe-stack-events` Perintah berikut menjelaskan peristiwa *myteststack* stack.

```
aws cloudformation describe-stack-events --stack-name myteststack
```

Berikut ini adalah contoh respons.

```
{
  "StackEvents": [
```

```

{
  "StackId": "arn:aws:cloudformation:aws-region:123456789012:stack/
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896",
  "EventId": "af67ef60-0b8f-11e3-8b8a-500150b352e0",
  "ResourceStatus": "CREATE_COMPLETE",
  "ResourceType": "AWS::CloudFormation::Stack",
  "Timestamp": "2013-08-23T01:02:30.070Z",
  "StackName": "myteststack",
  "PhysicalResourceId": "arn:aws:cloudformation:aws-
region:123456789012:stack/myteststack/a69442d0-0b8f-11e3-8b8a-500150b352e0",
  "LogicalResourceId": "myteststack"
},
{
  "StackId": "arn:aws:cloudformation:aws-region:123456789012:stack/
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896",
  "EventId": "S3Bucket-CREATE_COMPLETE-1377219748025",
  "ResourceStatus": "CREATE_COMPLETE",
  "ResourceType": "AWS::S3::Bucket",
  "Timestamp": "2013-08-23T01:02:28.025Z",
  "StackName": "myteststack",
  "ResourceProperties": "{\"AccessControl\": \"PublicRead\"}",
  "PhysicalResourceId": "myteststack-s3bucket-jssofilzie2w",
  "LogicalResourceId": "S3Bucket"
},
{
  "StackId": "arn:aws:cloudformation:aws-region:123456789012:stack/
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896",
  "EventId": "S3Bucket-CREATE_IN_PROGRESS-1377219746688",
  "ResourceStatus": "CREATE_IN_PROGRESS",
  "ResourceType": "AWS::S3::Bucket",
  "Timestamp": "2013-08-23T01:02:26.688Z",
  "ResourceStatusReason": "Resource creation Initiated",
  "StackName": "myteststack",
  "ResourceProperties": "{\"AccessControl\": \"PublicRead\"}",
  "PhysicalResourceId": "myteststack-s3bucket-jssofilzie2w",
  "LogicalResourceId": "S3Bucket"
},
{
  "StackId": "arn:aws:cloudformation:aws-region:123456789012:stack/
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896",
  "EventId": "S3Bucket-CREATE_IN_PROGRESS-1377219743862",
  "ResourceStatus": "CREATE_IN_PROGRESS",
  "ResourceType": "AWS::S3::Bucket",
  "Timestamp": "2013-08-23T01:02:23.862Z",

```

```
    "StackName": "myteststack",
    "ResourceProperties": "{\"AccessControl\": \"PublicRead\"}",
    "PhysicalResourceId": null,
    "LogicalResourceId": "S3Bucket"
  },
  {
    "StackId": "arn:aws:cloudformation:aws-region:123456789012:stack/myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896",
    "EventId": "a69469e0-0b8f-11e3-8b8a-500150b352e0",
    "ResourceStatus": "CREATE_IN_PROGRESS",
    "ResourceType": "AWS::CloudFormation::Stack",
    "Timestamp": "2013-08-23T01:02:15.422Z",
    "ResourceStatusReason": "User Initiated",
    "StackName": "myteststack",
    "PhysicalResourceId": "arn:aws:cloudformation:aws-region:123456789012:stack/myteststack/a69442d0-0b8f-11e3-8b8a-500150b352e0",
    "LogicalResourceId": "myteststack"
  }
]
```

Acara terbaru dilaporkan terlebih dahulu. Tabel berikut menjelaskan bidang yang dikembalikan oleh perintah `describe-stack-events`:

Bidang	Deskripsi
EventId	Pengidentifikasi acara.
StackName	Nama tumpukan yang sesuai dengan acara tersebut.
StackId	Pengidentifikasi tumpukan yang sesuai dengan acara tersebut.
LogicalResourceId	Pengidentifikasi logis sumber daya.
PhysicalResourceId	Pengidentifikasi fisik sumber daya.
ResourceProperties	Properti sumber daya.
ResourceType	Jenis sumber daya.
Timestamp	Waktu ketika peristiwa itu terjadi.

Bidang	Deskripsi
ResourceStatus	Status sumber daya, yang dapat menjadi salah satu kode status berikut: 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 DELETE_SKIPPED Status berlaku untuk sumber daya dengan atribut kebijakan penghapusan dari pertahankan.
DetailedStatus	Status rinci tumpukan. Jika CONFIGURATION_COMPLETE ada, fase konfigurasi sumber daya tumpukan telah selesai dan stabilisasi sumber daya sedang berlangsung.
ResourceStatusReason	Informasi lebih lanjut tentang status.

Kode status tumpukan

Tabel berikut menjelaskan kode status tumpukan:

Status tumpukan dan status rinci opsional	Deskripsi
CREATE_COMPLETE	Pembuatan satu atau lebih tumpukan berhasil.
CREATE_IN_PROGRESS	Pembuatan satu atau lebih tumpukan yang sedang berlangsung.

Status tumpukan dan status rinci opsional	Deskripsi
CREATE_FAILED	Pembuatan satu atau lebih tumpukan gagal. Lihat peristiwa tumpukan untuk melihat pesan kesalahan yang terkait. Kemungkinan alasan pembuatan yang gagal termasuk izin yang tidak memadai untuk bekerja dengan semua sumber daya di tumpukan, nilai parameter yang ditolak oleh AWS layanan, atau batas waktu selama pembuatan sumber daya.
DELETE_COMPLETE	Penghapusan satu atau lebih tumpukan berhasil. Tumpukan yang dihapus dipertahankan dan dapat dilihat selama 90 hari.
DELETE_FAILED	Penghapusan satu atau lebih tumpukan gagal. Karena penghapusan gagal, Anda mungkin memiliki beberapa sumber daya yang masih berjalan; tetapi, Anda tidak dapat menggunakan atau memperbarui tumpukan tersebut. Hapus tumpukan lagi atau lihat peristiwa tumpukan untuk melihat pesan kesalahan yang terkait.
DELETE_IN_PROGRESS	Penghapusan satu atau lebih tumpukan yang sedang berlangsung.
REVIEW_IN_PROGRESS	Pembuatan satu atau lebih tumpukan yang sedang berlangsung dengan StackId yang diharapkan tetapi tanpa templat atau sumber daya apa pun.  Important Tumpukan dengan kode status ini dihitung terhadap jumlah tumpukan maksimum yang memungkinkan.

Status tumpukan dan status rinci opsional	Deskripsi
ROLLBACK_COMPLETE	Penghapusan satu atau beberapa tumpukan berhasil setelah pembuatan tumpukan gagal atau setelah pembuatan tumpukan dibatalkan secara eksplisit. Tumpukan kembali ke status kerja sebelumnya. Semua sumber daya yang dibuat selama operasi pembuatan tumpukan akan dihapus. Status ini hanya ada setelah pembuatan tumpukan gagal. Ini menandakan bahwa semua operasi dari tumpukan yang dibuat sebagian telah dibersihkan dengan baik. Ketika dalam status ini, hanya operasi penghapusan yang dapat dilakukan.
ROLLBACK_FAILED	Penghapusan satu atau beberapa tumpukan gagal setelah pembuatan tumpukan gagal atau setelah pembuatan tumpukan dibatalkan secara eksplisit. Hapus tumpukan atau lihat peristiwa tumpukan untuk melihat pesan kesalahan yang terkait.
ROLLBACK_IN_PROGRESS	Penghapusan satu atau beberapa tumpukan yang sedang berlangsung setelah pembuatan tumpukan yang gagal atau setelah pembuatan tumpukan yang dibatalkan secara eksplisit .
UPDATE_COMPLETE	Pembaruan satu atau lebih tumpukan berhasil.
UPDATE_COMPLETE_CLEANUP_IN_PROGRESS	Penghapusan sumber daya lama yang sedang berlangsung untuk satu atau beberapa tumpukan setelah pembaruan tumpukan berhasil. Untuk pembaruan tumpukan yang membutuhkan sumber daya untuk diganti, CloudFormation buat sumber daya baru terlebih dahulu dan kemudian hapus sumber daya lama untuk membantu mengurangi gangguan apa pun dengan tumpukan Anda. Dalam keadaan ini, tumpukan telah diperbarui dan dapat digunakan, tetapi CloudFormation masih menghapus sumber daya lama.

Status tumpukan dan status rinci opsional	Deskripsi
UPDATE_FAILED	Pembaruan satu atau lebih tumpukan gagal. Lihat peristiwa tumpukan untuk melihat pesan kesalahan yang terkait.
UPDATE_IN_PROGRESS	Pembaruan dari satu atau lebih tumpukan yang sedang berlangsung.
UPDATE_ROLLBACK_COMPLETE	Berhasil mengembalikan satu atau beberapa tumpukan ke status kerja sebelumnya setelah pembaruan tumpukan gagal.
UPDATE_ROLLBACK_COMPLETE_CLEANUP_IN_PROGRESS	Penghapusan sumber daya baru yang sedang berlangsung untuk satu atau lebih tumpukan setelah pembaruan tumpukan gagal. Dalam keadaan ini, tumpukan telah digulung kembali ke status kerja sebelumnya dan dapat digunakan, tetapi CloudFormation masih menghapus sumber daya baru yang dibuatnya selama pembaruan tumpukan.
UPDATE_ROLLBACK_FAILED	Gagal mengembalikan satu atau beberapa tumpukan ke status kerja sebelumnya setelah pembaruan tumpukan gagal. Saat dalam status ini, Anda dapat menghapus tumpukan atau melanjutkan rollback . Anda mungkin perlu memperbaiki kesalahan sebelum tumpukan Anda dapat kembali ke status kerja. Atau, Anda dapat menghubungi Dukungan untuk mengembalikan tumpukan ke status yang dapat digunakan.
UPDATE_ROLLBACK_IN_PROGRESS	Pengembalian yang sedang berlangsung dari satu atau lebih tumpukan ke status kerja sebelumnya setelah pembaruan tumpukan gagal.
IMPORT_IN_PROGRESS	Operasi impor sedang berlangsung.
IMPORT_COMPLETE	Operasi impor telah berhasil diselesaikan untuk semua sumber daya di tumpukan yang mendukung <code>resource import</code> .

Status tumpukan dan status rinci opsional	Deskripsi
IMPORT_ROLLBACK_IN_PROGRESS	Impor akan kembali ke konfigurasi templat sebelumnya.
IMPORT_ROLLBACK_FAILED	Operasi rollback impor gagal untuk setidaknya satu sumber daya di tumpukan. Hasil akan tersedia untuk sumber daya yang CloudFormation berhasil diimpor.
IMPORT_ROLLBACK_COMPLETE	Impor berhasil dikembalikan ke konfigurasi templat sebelumnya.

Melihat garis waktu penerapan CloudFormation tumpukan

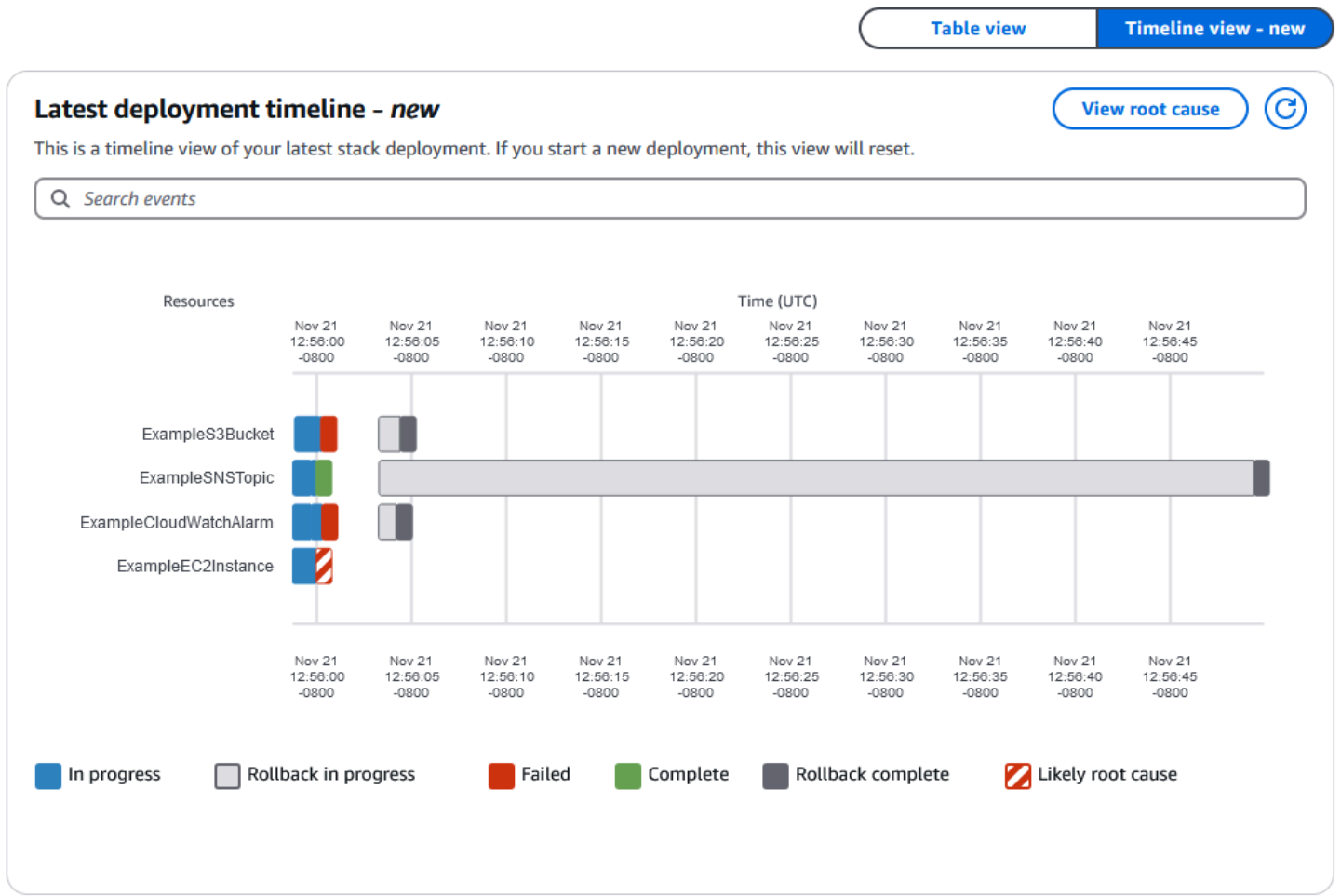
Grafik timeline penyebaran tumpukan menyediakan representasi visual dari timeline penyebaran tumpukan. Tampilan ini menunjukkan status penerapan untuk tumpukan dan setiap sumber dayanya, dan waktu setiap status berubah. Status penyebaran tumpukan diwakili oleh warna yang sesuai.

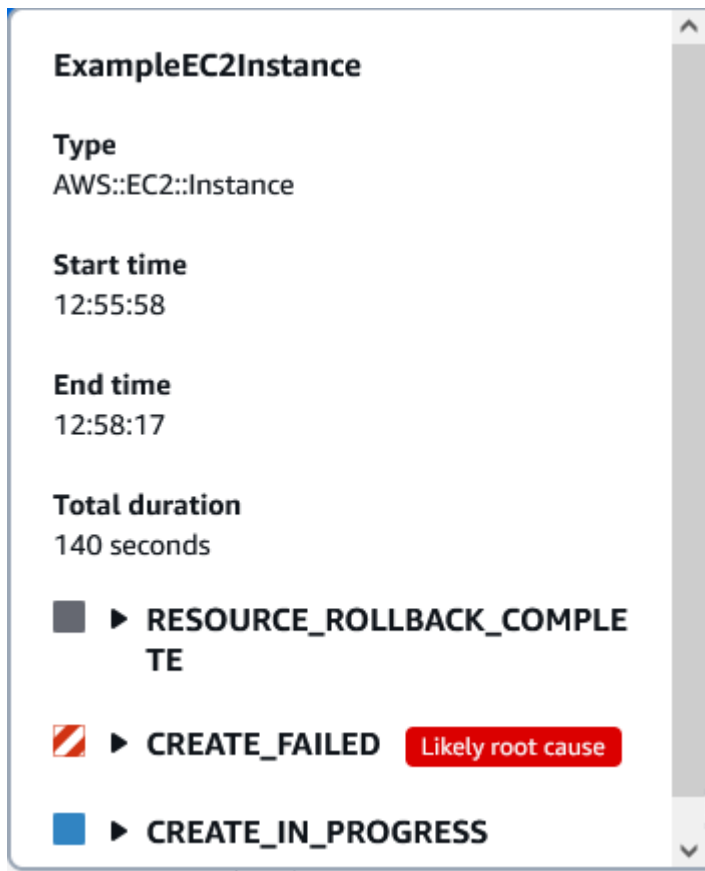
Topik

- [Memahami grafik timeline penerapan tumpukan](#)
- [Melihat grafik timeline penerapan tumpukan \(konsol\)](#)

Memahami grafik timeline penerapan tumpukan

Gambar berikut menunjukkan grafik timeline untuk penerapan tumpukan yang gagal karena sumber daya EC2 instans Amazon yang gagal diluncurkan.





Memilih sumber daya menunjukkan Jenis, waktu mulai penerapan, Waktu akhir penerapan, dan Durasi total penerapan. Anda juga akan menemukan Waktu mulai, Waktu akhir, dan Durasi setiap status penerapan di menu tarik-turun di bawah ini. Jika sumber daya gagal digunakan, alasan Kegagalan akan diberikan.

Untuk informasi selengkapnya tentang status tumpukan, lihat [Kode status tumpukan](#).

Melihat grafik timeline penerapan tumpukan (konsol)

Untuk melihat grafik timeline penerapan tumpukan:

1. Buka AWS CloudFormation konsol di <https://console.aws.amazon.com/cloudformation>.
2. Pada bilah navigasi di bagian atas layar, pilih tumpukan yang Wilayah AWS Anda buat.
3. Pada halaman Stacks CloudFormation konsol, pilih nama tumpukan. CloudFormation menampilkan detail tumpukan untuk tumpukan yang dipilih.
4. Pilih tab Acara untuk melihat peristiwa tumpukan yang CloudFormation telah dihasilkan untuk tumpukan Anda.
5. Pilih tombol Timeline graph untuk melihat grafik timeline untuk tumpukan Anda.

Memahami peristiwa pembuatan CloudFormation tumpukan

Selama penyebaran tumpukan, beberapa peristiwa terjadi untuk membuat, mengonfigurasi, dan memvalidasi sumber daya yang ditentukan dalam template tumpukan. Memahami peristiwa ini dapat membantu Anda mengoptimalkan proses pembuatan tumpukan dan merampingkan penerapan.

- Peristiwa pembuatan sumber daya - Saat setiap sumber daya memulai proses pembuatan, Status `CREATE_IN_PROGRESS` acara ditetapkan. Peristiwa ini menunjukkan bahwa sumber daya sedang disediakan.
- Pemeriksaan konsistensi akhir — Sebagian besar waktu pembuatan tumpukan dihabiskan untuk melakukan pemeriksaan konsistensi akhirnya terhadap sumber daya yang dibuat oleh tumpukan. Selama fase ini, layanan melakukan pemeriksaan konsistensi internal, memastikan sumber daya beroperasi penuh dan memenuhi kriteria stabilisasi layanan yang ditentukan oleh masing-masing Layanan AWS.
- Peristiwa lengkap konfigurasi - Ketika setiap sumber daya telah menyelesaikan fase pemeriksaan konsistensi akhirnya dari penyediaan, status Detail `CONFIGURATION_COMPLETE` peristiwa ditetapkan.
- Acara lengkap pembuatan sumber daya - Setelah sumber daya dibuat dan dikonfigurasi seperti yang ditentukan, dan konfigurasi cocok dengan apa yang ditentukan dalam templat, Status `CREATE_COMPLETE` acara diatur.

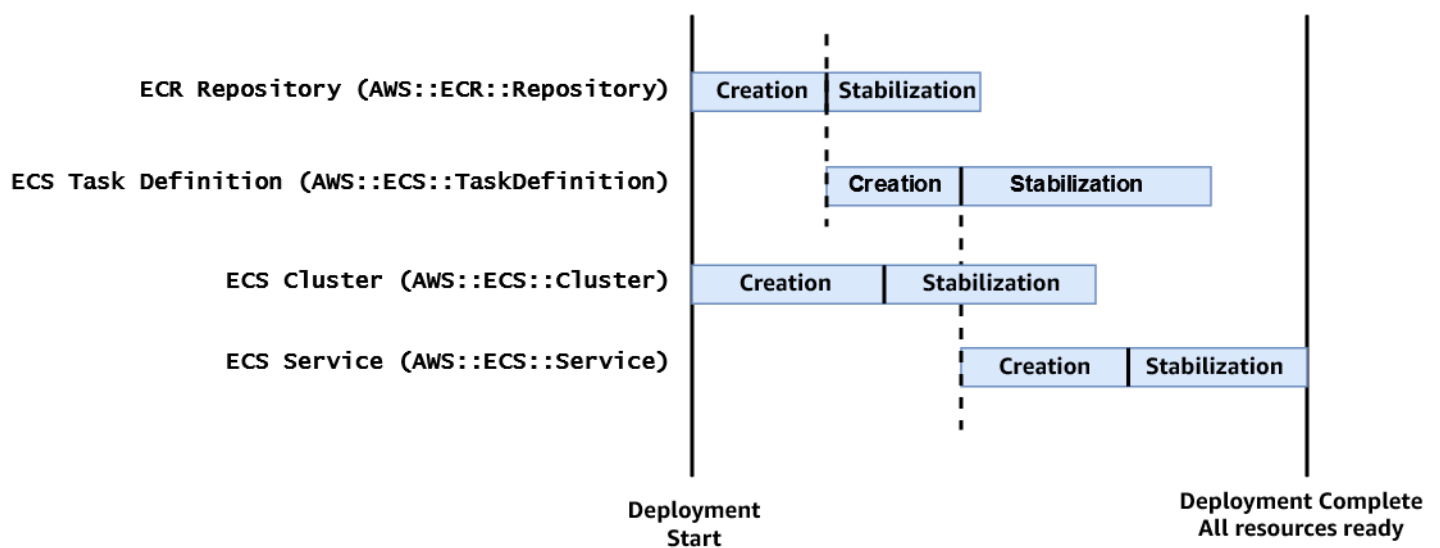
Anda dapat memanfaatkan `CONFIGURATION_COMPLETE` acara untuk merampingkan proses pembuatan tumpukan Anda dalam skenario di mana pemeriksaan konsistensi sumber daya akhirnya tidak diperlukan, seperti memvalidasi konfigurasi tumpukan pra-produksi atau penyediaan lintas-tumpukan. Anda dapat menggunakan acara ini dengan berbagai cara. Misalnya, Anda dapat menggunakannya sebagai sinyal visual untuk melewati menunggu sumber daya atau pemeriksaan konsistensi tumpukan selesai. Atau Anda dapat menggunakannya untuk membuat mekanisme otomatis menggunakan integrasi berkelanjutan dan pengiriman berkelanjutan (CI/CD) untuk memicu tindakan tambahan.

Important

Saat memanfaatkan `CONFIGURATION_COMPLETE` acara mempercepat waktu pembuatan tumpukan, Anda harus menyadari trade-offnya. Pertama, ini hanya didukung untuk subset tipe sumber daya yang mendukung deteksi drift. Untuk daftar jenis sumber daya yang mendukung deteksi drift, lihat [Dukungan jenis sumber daya](#). Pendekatan ini mungkin tidak

cocok untuk semua skenario, terutama di mana sumber daya memerlukan pemeriksaan konsistensi akhir yang menyeluruh untuk memastikan kesiapan operasional penuh di seluruh lingkungan cloud (misalnya, di lingkungan produksi). Sebaiknya hati-hati menilai persyaratan penerapan Anda dan kekritisan pemeriksaan konsistensi untuk setiap sumber daya. Gunakan `CONFIGURATION_COMPLETE` acara untuk mengoptimalkan kecepatan penerapan tanpa mengorbankan integritas dan keandalan infrastruktur Anda.

Karena `CONFIGURATION_COMPLETE` acara tidak dijamin akan disetel, skenario apa pun yang menggunakannya harus disiapkan untuk menangani `CREATE_COMPLETE` peristiwa ketika tidak ada `CONFIGURATION_COMPLETE` acara yang ditetapkan.



Saat penerapan tumpukan dimulai, baik `AWS::ECS::Cluster` sumber daya `AWS::ECR::Repository` maupun sumber daya memulai proses pembuatan (`ResourceStatus=CREATE_IN_PROGRESS`). Ketika tipe `AWS::ECR::Repository` sumber daya telah memulai pemeriksaan konsistensi akhirnya (`DetailedStatus=CONFIGURATION_COMPLETE`), maka `AWS::ECS::TaskDefinition` sumber daya dapat memulai proses pembuatan. Demikian pula, setelah `AWS::ECS::TaskDefinition` sumber daya memulai pemeriksaan konsistensi akhirnya, `AWS::ECS::Service` sumber daya memulai proses pembuatan.

CREATE_IN_PROGRESS dan **CREATE_COMPLETE** acara

- [Tumpukan]: `CREATE_IN_PROGRESS`
- [Sumber Daya]: Repositori ECR `CREATE_IN_PROGRESS`
- [Sumber Daya]: ECS Cluster `CREATE_IN_PROGRESS`

- [Sumber Daya]: **CREATE_IN_PROGRESS** Repositori ECR, CONFIGURATION_COMPLETE
- [Sumber Daya]: Definisi Tugas ECS CREATE_IN_PROGRESS
- [Sumber Daya]: ECS Cluster CREATE_IN_PROGRESS, CONFIGURATION_COMPLETE
- [Sumber Daya]: Definisi CREATE_IN_PROGRESS Tugas ECS, CONFIGURATION_COMPLETE
- [Sumber Daya]: Layanan ECS CREATE_IN_PROGRESS
- [Sumber Daya]: Repositori ECR CREATE_COMPLETE
- [Sumber Daya]: ECS Cluster CREATE_COMPLETE
- [Sumber Daya]: Layanan CREATE_IN_PROGRESS ECS, CONFIGURATION_COMPLETE
- [Tumpukan]: CREATE_IN_PROGRESS, CONFIGURATION_COMPLETE
- [Sumber Daya]: Definisi Tugas ECS CREATE_COMPLETE
- [Sumber Daya]: Layanan ECS CREATE_COMPLETE
- [Tumpukan]: CREATE_COMPLETE

Pantau kemajuan pembaruan tumpukan

Anda dapat memantau kemajuan pembaruan tumpukan dengan melihat kejadian tumpukan. Tab Peristiwa tumpukan menampilkan setiap langkah utama dalam pembuatan dan pembaruan tumpukan yang diurutkan berdasarkan waktu setiap acara dengan peristiwa terbaru di atasnya. Untuk informasi selengkapnya, lihat [Pantau kemajuan tumpukan](#).

Topik

- [Kejadian yang dihasilkan selama pembaruan tumpukan yang berhasil](#)
- [Kejadian yang dihasilkan ketika pembaruan sumber daya gagal](#)

Kejadian yang dihasilkan selama pembaruan tumpukan yang berhasil

Awal proses pembaruan tumpukan ditandai dengan UPDATE_IN_PROGRESS peristiwa untuk tumpukan:

```
2011-09-30 09:35 PDT AWS::CloudFormation::Stack MyStack UPDATE_IN_PROGRESS
```

Berikutnya adalah kejadian yang menandai awal dan penyelesaian pembaruan setiap sumber daya yang diubah dalam templat pembaruan. Misalnya, memperbarui [AWS::RDS::DBInstance](#) sumber daya bernama MyDB akan menghasilkan entri berikut:

```
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
```

UPDATE_IN_PROGRESSAcara dicatat ketika CloudFormation melaporkan bahwa ia telah mulai memperbarui sumber daya. Peristiwa UPDATE_COMPLETE dicatat ketika sumber daya berhasil dibuat.

Ketika CloudFormation telah berhasil memperbarui tumpukan, Anda akan melihat peristiwa berikut:

```
2011-09-30 09:35 PDT AWS::CloudFormation::Stack MyStack UPDATE_COMPLETE
```

Important

Selama operasi pembaruan tumpukan, jika CloudFormation perlu mengganti sumber daya yang ada, pertama-tama menciptakan sumber daya baru dan kemudian menghapus sumber daya lama. Namun, mungkin ada kasus di mana tidak CloudFormation dapat menghapus sumber daya lama (misalnya, jika pengguna tidak memiliki izin untuk menghapus sumber daya dari jenis tertentu).

CloudFormation melakukan tiga upaya untuk menghapus sumber daya lama. Jika tidak CloudFormation dapat menghapus sumber daya lama, itu menghapus sumber daya lama dari tumpukan dan terus memperbarui tumpukan. Ketika pembaruan tumpukan selesai, CloudFormation mengeluarkan peristiwa UPDATE_COMPLETE tumpukan, tetapi menyertakan StatusReason yang menyatakan bahwa satu atau beberapa sumber daya tidak dapat dihapus. CloudFormation juga mengeluarkan DELETE_FAILED acara untuk sumber daya tertentu, dengan yang sesuai StatusReason memberikan detail lebih lanjut tentang mengapa CloudFormation gagal menghapus sumber daya.

Sumber daya lama masih ada dan akan terus dikenakan biaya, tetapi tidak lagi dapat diakses melalui CloudFormation. Untuk menghapus sumber daya lama, akses sumber daya lama langsung menggunakan konsol atau API untuk layanan yang mendasarinya.

Hal ini juga berlaku untuk sumber daya yang telah Anda hapus dari tumpukan templat, dan akan dihapus dari tumpukan selama pembaruan tumpukan.

Kejadian yang dihasilkan ketika pembaruan sumber daya gagal

Jika pembaruan sumber daya gagal, CloudFormation laporkan `UPDATE_FAILED` peristiwa yang menyertakan alasan kegagalan. Misalnya, jika templat pembaruan Anda menetapkan perubahan properti yang tidak didukung oleh sumber daya seperti mengurangi ukuran `AllocatedStorage` [AWS::RDS::DBInstance](#) sumber daya, Anda akan melihat peristiwa seperti ini:

```
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
```

Jika pembaruan sumber daya gagal, mengembalikan CloudFormation sumber daya apa pun yang telah diperbarui selama pemutakhiran ke konfigurasinya sebelum pembaruan. Berikut adalah contoh kejadian yang akan Anda lihat selama rollback pembaruan:

```
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]
```

Terus mengembalikan pembaruan

Terkadang, ketika CloudFormation mencoba memutar kembali pembaruan tumpukan, itu tidak dapat memutar kembali semua perubahan yang dibuatnya selama proses pembaruan. Ini disebut `UPDATE_ROLLBACK_FAILED` negara. Misalnya, Anda mungkin memiliki tumpukan yang mulai memutar kembali ke instance database lama yang dihapus di luar CloudFormation. Karena CloudFormation tidak tahu basis data tersebut telah dihapus, itu mengasumsikan bahwa instans basis data tersebut masih ada dan mencoba untuk mengembalikan ke basis data tersebut, menyebabkan kegagalan pembaruan rollback.

Tumpukan dalam `UPDATE_ROLLBACK_FAILED` status tidak dapat diperbarui, tetapi dapat digulung kembali ke status kerja (`UPDATE_ROLLBACK_COMPLETE`). Setelah mengembalikan tumpukan ke pengaturan aslinya, Anda dapat mencoba memperbaruinya lagi.

Dalam sebagian besar kasus, Anda harus memperbaiki kesalahan yang menyebabkan pembaruan rollback gagal sebelum dapat melanjutkan untuk mengembalikan tumpukan Anda. Dalam kasus

lain, Anda dapat terus mengembalikan pembaruan tanpa perubahan apa pun, misalnya saat waktu operasi tumpukan habis.

 Note

Jika Anda menggunakan tumpukan bersarang, mengembalikan tumpukan induk akan mencoba untuk mengembalikan semua tumpukan anak juga.

Untuk terus mengembalikan pembaruan (konsol)

1. Masuk ke AWS Management Console dan buka AWS CloudFormation konsol di <https://console.aws.amazon.com/cloudformation>.
2. Pada bilah navigasi di bagian atas layar, pilih Wilayah AWS tempat tumpukan berada.
3. Pada halaman Stacks, pilih stack yang ingin Anda update, pilih Stack actions, dan kemudian pilih Continue update rollback.

Jika tidak ada solusi yang [Memecahkan masalah kesalahan](#) berhasil, Anda dapat menggunakan opsi lanjutan untuk melewati sumber daya yang CloudFormation tidak berhasil diputar kembali. Anda harus [mencari dan mengetik](#) logis IDs sumber daya yang ingin Anda lewati. Tentukan hanya sumber daya yang masuk ke UPDATE_FAILED status selama UpdateRollback dan tidak selama pembaruan penerusan.

 Warning

CloudFormation menetapkan status sumber daya yang ditentukan ke UPDATE_COMPLETE dan terus memutar kembali tumpukan. Setelah rollback selesai, status sumber daya yang dilewati akan tidak konsisten dengan status sumber daya di templat tumpukan. Sebelum Anda melakukan pembaruan tumpukan lain, Anda harus memperbarui tumpukan atau sumber daya agar konsisten satu sama lain. Jika tidak, pembaruan tumpukan berikutnya mungkin akan gagal dan tumpukan tidak akan dapat dipulihkan.

Tentukan jumlah minimum sumber daya yang diperlukan untuk berhasil mengembalikan tumpukan Anda. Sebagai contoh, pembaruan sumber daya yang gagal dapat menyebabkan kegagalan sumber daya dependen. Dalam hal ini, mungkin tidak perlu untuk melewati sumber daya dependen.

Untuk melewati sumber daya yang merupakan bagian dari tumpukan bersarang, gunakan format berikut: *NestedStackName.ResourceLogicalID*. Jika Anda ingin menentukan ID logis dari sumber daya tumpukan (Type: `AWS::CloudFormation::Stack`) di daftar `ResourcesToSkip`, tumpukan tertanam yang terkait harus berada dalam salah satu status berikut: `DELETE_IN_PROGRESS`, `DELETE_COMPLETE`, atau `DELETE_FAILED`.

Untuk terus mengembalikan pembaruan (AWS CLI)

- Gunakan [continue-update-rollback](#) perintah dengan `--stack-name` opsi untuk menentukan ID tumpukan yang ingin Anda lanjutkan untuk memutar kembali.

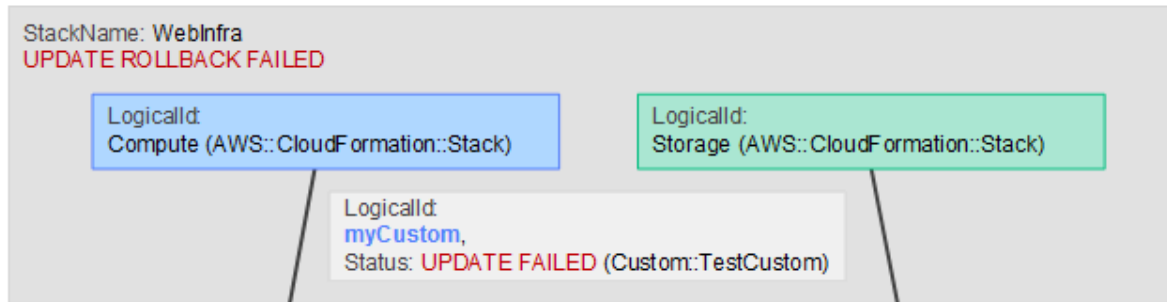
Lanjutkan memutar kembali dari pembaruan tumpukan bersarang yang gagal

Ketika Anda memiliki beberapa tumpukan yang bersarang satu sama lain, Anda mungkin perlu melewati sumber daya di beberapa level bersarang untuk mendapatkan hierarki tumpukan penuh kembali ke status kerja.

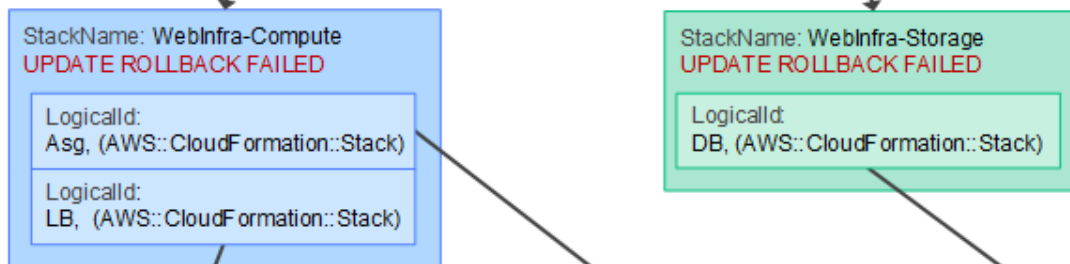
Misalnya, Anda memiliki tumpukan root yang disebut `WebInfra` yang berisi dua tumpukan kecil di dalamnya: `WebInfra-Compute` dan `WebInfra-Storage`. Kedua tumpukan ini juga memiliki tumpukan bersarang sendiri di dalamnya.

Jika terjadi kesalahan selama pembaruan, dan proses pembaruan gagal, seluruh hierarki tumpukan mungkin berakhir dalam `UPDATE_ROLLBACK_FAILED` status, seperti yang ditunjukkan pada diagram berikut.

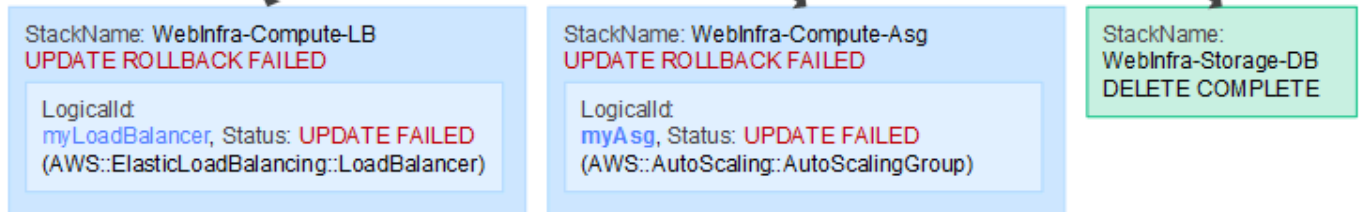
Root



Level 1



Level 2

**Note**

Nama tumpukan dalam contoh ini dipotong agar memudahkan. Nama tumpukan anak biasanya dihasilkan oleh CloudFormation dan berisi string acak yang unik, jadi nama sebenarnya mungkin tidak ramah pengguna.

Untuk mendapatkan tumpukan root ke status yang dapat dioperasikan menggunakan `continue-update-rollback` perintah, Anda harus menggunakan `--resources-to-skip` opsi untuk melewati sumber daya yang gagal di-rollback.

`continue-update-rollback` Contoh berikut melanjutkan operasi rollback dari pembaruan tumpukan yang sebelumnya gagal. Dalam contoh ini, `--resources-to-skip` opsi mencakup item berikut:

- *myCustom*
- *WebInfra-Compute-Asg.myAsg*
- *WebInfra-Compute-LB.myLoadBalancer*
- *WebInfra-Storage.DB*

Untuk sumber daya tumpukan root, Anda hanya perlu memberikan ID logis, misalnya, *myCustom*. Namun, untuk sumber daya yang terkandung dalam tumpukan bersarang, Anda harus memberikan nama tumpukan bersarang dan ID logisnya, dipisahkan oleh titik. Misalnya, *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
```

Untuk menemukan nama tumpukan tumpukan bersarang

Anda dapat menemukannya di dalam ID tumpukan anak atau Nama Sumber Daya Amazon (ARN).

Contoh ARN berikut mengacu pada tumpukan bernama. *WebInfra-Storage-Z2VKC706XKXT*

```
arn:aws:cloudformation:us-east-1:123456789012:stack/WebInfra-Storage-Z2VKC706XKXT/  
ea9e7f90-54f7-11e6-a032-028f3d2330bd
```

Untuk menemukan ID logis dari tumpukan bersarang

Anda dapat menemukan ID logis tumpukan anak dalam definisi templat induknya. Dalam diagram, tumpukan *WebInfra-Storage-DB* anak ada DB di induknya *WebInfra-Storage*. *LogicalId*

Di CloudFormation konsol, Anda juga dapat menemukan ID logis di kolom Logical ID untuk sumber daya tumpukan pada tab Resources atau tab Events. Untuk informasi selengkapnya, lihat [Lihat informasi tumpukan dari CloudFormation konsol](#).

Tentukan penyebab kegagalan tumpukan

Jika pembuatan tumpukan Anda gagal, CloudFormation dapat membantu Anda menentukan peristiwa yang kemungkinan merupakan akar penyebab kegagalan tumpukan. Bergantung pada skenario dan izin Anda, AWS CloudTrail peristiwa mungkin dapat memberikan rincian lebih lanjut tentang akar penyebab jika alasan Status yang disediakan di Acara tidak jelas.

Untuk menentukan akar penyebab kegagalan tumpukan

1. Buka AWS CloudFormation konsol di <https://console.aws.amazon.com/cloudformation>.
2. Pada halaman Stacks, pilih tumpukan gagal.
3. Pilih tab Acara.
4. Pilih Deteksi akar penyebab. CloudFormation akan menganalisis kegagalan dan menunjukkan peristiwa yang kemungkinan menjadi penyebab kegagalan dengan menambahkan label kemungkinan akar penyebab ke Status peristiwa tertentu. Lihat Alasan status untuk penjelasan lebih lanjut tentang status di CloudFormation konsol.
5. Pilih Status gagal dengan label Kemungkinan akar penyebab untuk mempelajari lebih lanjut tentang penyebab kegagalan. Bergantung pada skenario dan izin Anda, Anda mungkin dapat meninjau CloudTrail peristiwa terperinci. Ini adalah hasil potensial berikut dari memilih Status
 - CloudTrail peristiwa yang terkait dengan masalah ini tersedia dan dapat membantu penyelesaian. Lihat CloudTrail acara.
 - Kami tidak dapat menemukan CloudTrail peristiwa apa pun yang terkait dengan masalah ini yang dapat membantu penyelesaian.
 - Izin Anda saat ini tidak mengizinkan akses untuk melihat CloudTrail peristiwa. Pelajari selengkapnya.
 - Dalam proses memeriksa CloudTrail acara yang tersedia, periksa kembali dalam beberapa menit.
 - Terjadi kesalahan saat mengambil CloudTrail peristiwa. Untuk pemeriksaan manual, kunjungi CloudTrail konsol.
6. Jika alasan yang diberikan dalam alasan Status tidak jelas, dan akar penyebab menampilkan tautan ke CloudTrail konsol, buka tautan untuk melihat acara untuk menemukan akar penyebab terperinci.

Untuk informasi selengkapnya tentang CloudTrail peristiwa, lihat [Memahami CloudTrail peristiwa](#) dan [CloudTrail merekam konten](#).

Untuk informasi selengkapnya tentang riwayat CloudTrail acara, lihat [Bekerja dengan riwayat CloudTrail Acara](#).

 Note

Tumpukan bersarang tidak mendukung Deteksi akar penyebab.

Pilih cara menangani kegagalan saat menyediakan sumber daya

Jika operasi tumpukan Anda gagal, Anda tidak perlu memutar kembali sumber daya yang sudah berhasil disediakan dan memulai dari awal setiap saat. Sebagai gantinya, Anda dapat memecahkan masalah sumber daya dalam UPDATE_FAILED status CREATE_FAILED atau, lalu melanjutkan penyediaan dari titik di mana masalah terjadi.

Untuk melakukan ini, Anda harus mengaktifkan opsi mempertahankan sumber daya yang berhasil disediakan. Opsi ini tersedia untuk semua penerapan tumpukan dan operasi set perubahan.

- Untuk pembuatan tumpukan, jika Anda memilih opsi Pertahankan sumber daya yang berhasil disediakan, CloudFormation mempertahankan status sumber daya yang berhasil dibuat dan membiarkan yang gagal dalam status gagal hingga operasi pembaruan berikutnya dilakukan.
- Selama operasi set pembaruan dan perubahan, memilih Pertahankan sumber daya yang berhasil disediakan mempertahankan status sumber daya yang berhasil sambil mengembalikan sumber daya yang gagal ke status stabil terakhir yang diketahui. Sumber daya yang gagal akan berada dalam UPDATE_FAILED keadaan. Sumber daya tanpa status stabil terakhir yang diketahui akan dihapus pada operasi tumpukan berikutnya.

Topik

- [Ikhtisar opsi kegagalan tumpukan](#)
- [Kondisi yang diperlukan untuk menjeda stack rollback](#)
- [Pertahankan sumber daya yang berhasil disediakan \(konsol\)](#)
- [Pertahankan sumber daya yang berhasil disediakan \(AWS CLI\)](#)

Ikhtisar opsi kegagalan tumpukan

Sebelum mengeluarkan operasi dari CloudFormation konsol, API, atau AWS CLI, tentukan perilaku kegagalan sumber daya yang disediakan. Kemudian, lanjutkan dengan proses penyebaran sumber daya Anda tanpa modifikasi lainnya. Jika terjadi kegagalan operasional, CloudFormation berhenti

pada kegagalan pertama di setiap jalur penyediaan independen. CloudFormation mengidentifikasi dependensi antara sumber daya untuk memparalelkan tindakan penyediaan independen. Kemudian ia melanjutkan untuk menyediakan sumber daya pada setiap jalur penyediaan independen sampai mengalami kegagalan. Kegagalan di satu jalur tidak memengaruhi jalur penyediaan lainnya. CloudFormation akan terus menyediakan sumber daya sampai selesai atau berhenti pada kegagalan yang berbeda.

Memperbaiki masalah apa pun untuk melanjutkan proses penerapan. CloudFormation melakukan pembaruan yang diperlukan sebelum mencoba kembali tindakan penyediaan pada sumber daya yang tidak dapat berhasil disediakan sebelumnya. Anda memulihkan masalah dengan mengirimkan operasi Coba Ulang, Perbarui, atau Gulung kembali. Misalnya, jika Anda menyediakan instans Amazon dan EC2 instans gagal selama operasi pembuatan, Anda mungkin ingin menyelidiki kesalahan tersebut, daripada segera mengembalikan sumber daya yang gagal. EC2 Anda dapat meninjau pemeriksaan status sistem dan pemeriksaan status instance, lalu pilih operasi Coba lagi setelah masalah diselesaikan.

Ketika operasi tumpukan gagal, dan Anda telah menentukan Pertahankan sumber daya yang berhasil disediakan dari menu opsi kegagalan Stack, Anda dapat memilih opsi berikut.

- **Coba lagi** — Mencoba lagi operasi penyediaan pada sumber daya yang gagal dan terus menyediakan template hingga berhasil menyelesaikan operasi tumpukan atau kegagalan berikutnya. Pilih opsi ini jika sumber daya gagal disediakan karena masalah yang tidak memerlukan modifikasi templat, seperti izin AWS Identity and Access Management (IAM).
- **Update** - Sumber daya yang telah disediakan diperbarui pada pembaruan template. Sumber daya yang gagal dibuat atau diperbarui akan dicoba lagi. Pilih opsi ini jika sumber daya gagal disediakan karena kesalahan templat, dan Anda telah memodifikasi templat. Saat memperbarui tumpukan yang berada dalam FAILED status, Anda harus memilih Pertahankan sumber daya yang berhasil disediakan untuk opsi kegagalan Stack untuk terus memperbarui tumpukan Anda.
- **Gulung kembali** — CloudFormation memutar kembali tumpukan ke keadaan stabil terakhir yang diketahui.

Kondisi yang diperlukan untuk menjeda stack rollback

Untuk CloudFormation mencegah agar tidak secara otomatis memutar kembali dan menghapus sumber daya yang berhasil dibuat, kondisi berikut harus dipenuhi.

1. Saat Anda membuat atau memperbarui tumpukan, Anda harus memilih opsi untuk Mempertahankan sumber daya yang berhasil disediakan. Ini memberi tahu untuk CloudFormation

tidak menghapus sumber daya yang berhasil dibuat, bahkan jika operasi tumpukan keseluruhan gagal.

2. Operasi tumpukan pasti gagal, artinya status tumpukan adalah salah satu `CREATE_FAILED` atau `UPDATE_FAILED`.

 Note

Jenis pembaruan yang tidak dapat diubah tidak didukung.

Pertahankan sumber daya yang berhasil disediakan (konsol)

Create stack

Untuk mempertahankan sumber daya yang berhasil disediakan selama operasi create stack

1. Masuk ke AWS Management Console dan buka AWS CloudFormation konsol di <https://console.aws.amazon.com/cloudformation>.
2. Dari halaman Stacks, pilih Buat tumpukan di kanan atas, lalu pilih Dengan sumber daya baru (standar).
3. Untuk Prasyarat - Siapkan template, pilih Pilih template yang ada.
4. Dalam Tentukan templat, pilih untuk menentukan URL untuk bucket S3 yang berisi templat tumpukan Anda atau unggah file templat tumpukan. Lalu, pilih Selanjutnya.
5. Pada halaman Tentukan detail tumpukan, masukkan nama tumpukan di kotak Nama tumpukan.
6. Di bagian Parameter, tentukan parameter yang ditentukan dalam template tumpukan Anda.

Anda dapat menggunakan atau mengubah parameter dengan nilai default.

7. Bila Anda puas dengan nilai parameter, pilih Berikutnya.
8. Pada halaman Configure stack options, Anda dapat mengatur opsi tambahan untuk tumpukan Anda.
9. Untuk opsi kegagalan Stack, pilih Pertahankan sumber daya yang berhasil disediakan.
10. Ketika Anda puas dengan opsi tumpukan, pilih Berikutnya.
11. Tinjau tumpukan Anda di halaman Ulasan dan pilih Buat tumpukan.

Hasil: Sumber daya yang gagal membuat transisi status tumpukan `CREATE_FAILED` untuk mencegah tumpukan berputar kembali saat operasi tumpukan mengalami kegagalan. Sumber daya yang berhasil disediakan berada dalam keadaan `CREATE_COMPLETE`. Anda dapat memantau tumpukan di tab Stack events.

Update stack

Untuk mempertahankan sumber daya yang berhasil disediakan selama operasi tumpukan pembaruan

1. Masuk ke AWS Management Console dan buka AWS CloudFormation konsol di <https://console.aws.amazon.com/cloudformation>.
2. Pilih tumpukan yang ingin Anda perbarui dan kemudian pilih Perbarui.
3. Pada halaman Update stack, pilih template stack dengan menggunakan salah satu opsi berikut:
 - Gunakan template yang ada
 - Ganti template saat ini
 - Edit template di Infrastructure Composer

Terima pengaturan Anda dan pilih Berikutnya.

4. Pada halaman Tentukan detail tumpukan, tentukan parameter yang ditentukan dalam template tumpukan Anda.

Anda dapat menggunakan atau mengubah parameter dengan nilai default.

5. Bila Anda puas dengan nilai parameter, pilih Berikutnya.
6. Pada halaman Configure stack options, Anda dapat mengatur opsi tambahan untuk tumpukan Anda.
7. Untuk kegagalan perilaku penyediaan, pilih Pertahankan sumber daya yang berhasil disediakan.
8. Ketika Anda puas dengan opsi tumpukan, pilih Berikutnya.
9. Tinjau tumpukan Anda di halaman Ulasan dan pilih Perbarui tumpukan.

Hasil: Sumber daya yang gagal memperbarui mentransisikan status tumpukan ke `UPDATE_FAILED` dan memutar kembali ke status stabil terakhir yang diketahui. Sumber daya tanpa status stabil terakhir yang diketahui akan dihapus CloudFormation pada operasi tumpukan

berikutnya. Sumber daya yang berhasil disediakan berada dalam keadaan CREATE_COMPLETE atau UPDATE_COMPLETE. Anda dapat memantau tumpukan di tab Stack events.

Change set

Note

Anda dapat memulai set perubahan untuk tumpukan dengan status CREATE_FAILED atau UPDATE_FAILED, tetapi tidak untuk status. UPDATE_ROLLBACK_FAILED

Untuk Mempertahankan sumber daya yang berhasil disediakan selama operasi set perubahan

1. Masuk ke AWS Management Console dan buka AWS CloudFormation konsol di <https://console.aws.amazon.com/cloudformation>.
2. Pilih tumpukan yang berisi set perubahan yang ingin Anda mulai, lalu pilih tab Ubah set.
3. Pilih set perubahan dan kemudian pilih Execute.
4. Untuk Jalankan set perubahan, pilih opsi Pertahankan sumber daya yang berhasil disediakan.
5. Pilih Jalankan set perubahan.

Hasil: Sumber daya yang gagal memperbarui mentransisikan status tumpukan ke UPDATE_FAILED dan memutar kembali ke status stabil terakhir yang diketahui. Sumber daya tanpa status stabil terakhir yang diketahui akan dihapus CloudFormation pada operasi tumpukan berikutnya. Sumber daya yang berhasil disediakan berada dalam keadaan CREATE_COMPLETE atau UPDATE_COMPLETE. Anda dapat memantau tumpukan di tab Stack events.

Pertahankan sumber daya yang berhasil disediakan ()AWS CLI

Create stack

Untuk mempertahankan sumber daya yang berhasil disediakan selama operasi pembuatan tumpukan

[Tentukan --disable-rollback opsi atau on-failure DO_NOTHING enumerasi selama operasi create-stack.](#)

1. Berikan nama tumpukan dan templat ke create-stack perintah dengan `--disable-rollback` opsi.

```
aws cloudformation create-stack --stack-name myteststack \  
  --template-body file://template.yaml \  
  --disable-rollback
```

Perintah mengembalikan output berikut.

```
{  
  "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/  
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896"  
}
```

2. Jelaskan status tumpukan menggunakan describe-stacks perintah.

```
aws cloudformation describe-stacks --stack-name myteststack
```

Perintah mengembalikan output berikut.

```
{  
  "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

Untuk mempertahankan sumber daya yang berhasil disediakan selama operasi pembaruan tumpukan

1. Berikan nama tumpukan dan templat yang ada ke update-stack perintah dengan --disable-rollback opsi.

```
aws cloudformation update-stack --stack-name myteststack \  
  --template-url https://s3.amazonaws.com/amzn-s3-demo-bucket/updated.template \  
  --disable-rollback
```

Perintah mengembalikan output berikut.

```
{  
  "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/  
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896"  
}
```

2. Jelaskan status tumpukan menggunakan describe-stack-events perintah describe-stacks atau.

```
aws cloudformation describe-stacks --stack-name myteststack
```

Perintah mengembalikan output berikut.

```
{  
  "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

Anda dapat memulai set perubahan untuk tumpukan dengan status `CREATE_FAILED` atau `UPDATE_FAILED` tetapi tidak untuk status `UPDATE_ROLLBACK_FAILED`.

Untuk mempertahankan sumber daya yang berhasil disediakan selama operasi set perubahan

Tentukan `--disable-rollback` opsi selama [execute-change-set](#) operasi.

1. Berikan nama tumpukan dan templat ke `execute-change-set` perintah dengan `--disable-rollback` opsi.

```
aws cloudformation execute-change-set --stack-name myteststack \  
  --change-set-name my-change-set --template-body file://template.yaml
```

Perintah mengembalikan output berikut.

```
{  
  "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. Memulai set perubahan dengan `--disable-rollback` opsi.

```
aws cloudformation execute-change-set --stack-name myteststack \  
  --change-set-name my-change-set --disable-rollback
```

3. Tentukan status tumpukan menggunakan `describe-stack-events` perintah `describe-stacks` atau.

```
aws cloudformation describe-stack-events --stack-name myteststack
```

Perintah mengembalikan output berikut.

```
{
  "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-abcdefghijkl1",
      "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. Perbaiki kesalahan izin dan coba lagi operasi.

```
aws cloudformation update-stack --stack-name myteststack \
--use-previous-template --disable-rollback
```

Perintah mengembalikan output berikut.

```
{
  "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896"
}
```

5. Jelaskan status tumpukan menggunakan describe-stack-events perintah describe-stacks atau.

```
aws cloudformation describe-stacks --stack-name myteststack
```

Perintah mengembalikan output berikut.

```
{
  "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
}
```

Menggulung kembali tumpukan

Anda dapat menggunakan perintah [rollback-stack](#) untuk memutar kembali tumpukan dengan status tumpukan `CREATE_FAILED` atau `UPDATE_FAILED` tumpukan ke status stabil terakhirnya.

`rollback-stack` Perintah berikut memutar kembali tumpukan yang ditentukan.

```
aws cloudformation rollback-stack --stack-name myteststack
```

Perintah mengembalikan output berikut.

```
{
  "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896"
}
```

Note

`rollback-stack` Operasi akan menghapus tumpukan jika tidak berisi status stabil terakhir yang diketahui.

Gulung kembali CloudFormation tumpukan Anda pada pelanggaran alarm dengan pemicu rollback

Dengan pemicu rollback, Anda dapat AWS CloudFormation memantau status aplikasi Anda selama pembuatan dan pembaruan tumpukan, dan memutar kembali operasi itu jika aplikasi melanggar ambang alarm yang telah Anda tentukan. Untuk setiap pemicu rollback yang Anda buat, Anda menentukan CloudWatch alarm yang CloudFormation harus dipantau. CloudFormation memonitor alarm yang ditentukan selama operasi pembuatan atau pembaruan tumpukan, dan untuk jumlah waktu yang ditentukan setelah semua sumber daya digunakan. Jika salah satu alarm ALARM berstatus selama operasi tumpukan atau periode pemantauan, putar kembali CloudFormation seluruh operasi tumpukan.

Anda dapat mengatur waktu pemantauan dari default 0 hingga 180 menit. Selama waktu ini, CloudFormation memantau semua pemicu rollback setelah operasi pembuatan atau pembaruan tumpukan men-deploy semua sumber daya yang diperlukan. Jika salah satu alarm ALARM berstatus selama operasi tumpukan atau periode pemantauan ini, putar CloudFormation kembali seluruh operasi tumpukan. Kemudian, untuk operasi pembaruan, jika periode pemantauan berakhir tanpa ada alarm yang akan ALARM berstatus, CloudFormation lanjutkan untuk membuang sumber daya lama seperti biasa. Jika Anda menetapkan waktu pemantauan tetapi tidak menentukan pemicu rollback apa pun, CloudFormation tetap menunggu periode waktu yang ditentukan sebelum membersihkan sumber daya lama untuk operasi pembaruan. Anda dapat menggunakan periode pemantauan ini untuk melakukan validasi tumpukan manual yang diinginkan, dan membatalkan pembuatan atau pembaruan tumpukan secara manual jika diperlukan. Jika Anda mengatur waktu pemantauan 0 menit, CloudFormation masih memantau pemicu rollback selama operasi pembuatan dan pembaruan tumpukan dan mengembalikan operasi jika alarm memasuki status ALARM. Kemudian, untuk operasi pembaruan tanpa alarm yang melanggar, ini mulai membuang sumber daya lama segera setelah operasi selesai.

Secara default, CloudFormation hanya memutar kembali operasi tumpukan jika alarm masuk ke ALARM status, bukan INSUFFICIENT_DATA status. Untuk CloudFormation memutar kembali operasi tumpukan jika alarm juga INSUFFICIENT_DATA berstatus, edit CloudWatch alarm untuk memperlakukan data yang hilang sebagaibreaching. Untuk informasi selengkapnya, lihat [Mengonfigurasi cara CloudWatch alarm menangani data yang hilang](#) di CloudWatch Panduan Pengguna Amazon.

CloudFormation tidak memantau pemicu rollback saat memutar kembali tumpukan selama operasi pembaruan.

Anda dapat menambahkan maksimal 5 pemicu rollback. Untuk menambahkan pemicu rollback, tentukan Nama Sumber Daya Amazon (ARN) alarm. CloudWatch Saat ini, `AWS::CloudWatch::Alarm` dan `AWS::CloudWatch::CompositeAlarm` tipe dapat digunakan sebagai pemicu rollback. Untuk informasi selengkapnya tentang CloudWatch alarm, lihat [Menggunakan CloudWatch alarm](#) di CloudWatch Panduan Pengguna Amazon.

Jika CloudWatch alarm yang diberikan hilang, seluruh operasi tumpukan gagal dan berputar kembali.

Ketahui bahwa akses ke CloudWatch memerlukan kredensi. Kredensi tersebut harus memiliki izin untuk mengakses AWS sumber daya, seperti mengambil data CloudWatch metrik tentang sumber daya Anda. Untuk informasi selengkapnya, lihat [Otentikasi dan kontrol akses CloudWatch](#) di Panduan CloudWatch Pengguna Amazon.

Tambahkan pemicu rollback selama pembuatan tumpukan atau pembaruan

Untuk menambahkan pemicu rollback selama pembuatan atau pembaruan tumpukan (konsol)

1. Saat membuat atau memperbarui tumpukan, pada halaman Configure stack options, di bawah Advanced options, perluas bagian konfigurasi Rollback.
2. Masukkan waktu pemantauan antara 0 - 180 menit. Nilai default-nya adalah 0.
3. Tentukan ARN CloudWatch alarm atau alarm komposit yang ingin Anda gunakan sebagai pemicu rollback, dan pilih Add alarm ARN. CloudWatch

Misalnya, berikut ini adalah ARN untuk CloudWatch alarm atau alarm komposit,.

`arn:aws:cloudwatch:us-east-1:123456789012:alarm:MyAlarmName`

4. Pilih Berikutnya dan tinjau detail tumpukan Anda.
5. Saat Anda siap, pilih Kirim untuk membuat atau memperbarui tumpukan.

Untuk menambahkan pemicu rollback selama pembuatan stack atau update ()AWS CLI

Gunakan [create-stack](#) atau [update-stack](#) perintah dengan `--rollback-configuration` opsi.

Misalnya, `update-stack` perintah berikut ditetapkan `MyCompositeAlarm` sebagai pemicu rollback dengan periode pemantauan 5 menit:

```
aws cloudformation update-stack --stack-name MyStack \  
  --use-previous-template \  
  --rollback-configuration \  
    --alarm-arn arn:aws:cloudwatch:us-east-1:123456789012:alarm:MyCompositeAlarm
```



```
"RollbackTriggers=[{Arn=arn:aws:cloudwatch:us-east-1:123456789012:alarm:MyCompositeAlarm,Type=AWS::CloudWatch::CompositeAlarm}],MonitoringTim
```

Tambahkan pemicu rollback ke set perubahan

Untuk menambahkan pemicu rollback ke set perubahan (konsol)

1. Saat membuat set perubahan, pada halaman Configure stack options, di bawah Advanced options, perluas bagian konfigurasi Rollback.
2. Masukkan waktu pemantauan antara 0 - 180 menit. Nilai default-nya adalah 5.
3. Tentukan ARN CloudWatch alarm atau alarm komposit yang ingin Anda gunakan sebagai pemicu rollback, dan pilih Add alarm ARN. CloudWatch

Misalnya, berikut ini adalah ARN untuk CloudWatch alarm atau alarm komposit,.
arn:aws:cloudwatch:us-east-1:123456789012:alarm:MyAlarmName

4. Pilih Berikutnya dan tinjau detail set perubahan Anda.
5. Saat Anda siap, pilih Buat set perubahan untuk membuat set perubahan.

Untuk menambahkan pemicu rollback ke set perubahan (AWS CLI)

Gunakan [create-change-set](#) perintah dengan --rollback-configuration opsi.

Lihat pemicu rollback untuk tumpukan

Untuk melihat pemicu rollback untuk tumpukan, lihat bagian konfigurasi Rollback.

1. Pada halaman Stacks, pilih tumpukan yang ingin Anda lihat dari daftar di sebelah kiri.
2. Pada tab Stack info, perluas bagian konfigurasi Rollback untuk melihat pemicu rollback.

Lihat pemicu rollback untuk set perubahan

Untuk melihat pemicu rollback untuk set perubahan, lihat bagian konfigurasi Rollback.

1. Pada halaman Stacks, pilih tumpukan yang ingin Anda lihat dari daftar di sebelah kiri.
2. Pilih tab Ubah set, lalu pilih set perubahan yang ingin Anda lihat.
3. Pilih tab Input, dan lihat bagian konfigurasi Rollback.

Mendeteksi perubahan konfigurasi yang tidak terkelola pada tumpukan dan sumber daya dengan deteksi drift

Bahkan saat Anda mengelola sumber daya Anda CloudFormation, pengguna dapat mengubah sumber daya tersebut di luar CloudFormation. Pengguna dapat mengedit sumber daya secara langsung dengan menggunakan layanan dasar yang membuat sumber daya. Misalnya, Anda dapat menggunakan EC2 konsol Amazon untuk memperbarui instance server yang dibuat sebagai bagian dari CloudFormation tumpukan. Beberapa perubahan mungkin tidak disengaja, dan beberapa mungkin dibuat dengan sengaja untuk menanggapi peristiwa operasional yang sensitif terhadap waktu. Terlepas dari itu, perubahan yang dilakukan di luar CloudFormation dapat mempersulit pembaruan tumpukan atau operasi penghapusan. Anda dapat menggunakan deteksi drift untuk mengidentifikasi sumber daya tumpukan yang telah dilakukan perubahan konfigurasi di luar CloudFormation manajemen. Anda kemudian dapat mengambil tindakan korektif sehingga sumber daya tumpukan Anda kembali sinkron dengan definisinya di templat tumpukan, seperti memperbarui sumber daya menyimpang secara langsung sehingga mereka setuju dengan definisi templatnya. Menyelesaikan penyimpangan membantu memastikan konsistensi konfigurasi dan operasi tumpukan yang sukses.

Topik

- [Apa yang dimaksud dengan penyimpangan?](#)
- [Kode status deteksi penyimpangan](#)
- [Pertimbangan saat mendeteksi penyimpangan](#)
- [Mendeteksi penyimpangan di seluruh tumpukan CloudFormation](#)
- [Mendeteksi penyimpangan di sumber daya tumpukan individu](#)
- [Mengatasi penyimpangan dengan operasi impor](#)

Apa yang dimaksud dengan penyimpangan?

Deteksi penyimpangan memungkinkan Anda mendeteksi apakah konfigurasi sebenarnya dari suatu tumpukan berbeda, atau telah menyimpang, dari konfigurasi yang diharapkan. Gunakan CloudFormation untuk mendeteksi penyimpangan pada seluruh tumpukan, atau pada sumber daya individu dalam tumpukan. Sumber daya dianggap telah menyimpang jika salah satu nilai properti sebenarnya berbeda dari nilai properti yang diharapkan. Ini termasuk jika properti atau sumber daya telah dihapus. Tumpukan dianggap telah menyimpang jika satu atau lebih dari sumber dayanya menyimpang.

Untuk menentukan apakah sumber daya telah hanyut, CloudFormation tentukan nilai properti sumber daya yang diharapkan, seperti yang didefinisikan dalam template tumpukan dan nilai apa pun yang ditentukan sebagai parameter templat. CloudFormation kemudian membandingkan nilai-nilai yang diharapkan dengan nilai aktual dari properti sumber daya seperti yang saat ini ada di tumpukan. Sebuah sumber daya dianggap telah menyimpang jika satu atau lebih dari propertinya telah dihapus, atau nilainya diubah.

CloudFormation menghasilkan informasi rinci tentang setiap sumber daya di tumpukan yang telah melayang.

CloudFormation mendeteksi penyimpangan pada AWS sumber daya yang mendukung deteksi drift. Sumber daya yang tidak mendukung deteksi penyimpangan diberi status penyimpangan NOT_CHECKED. Untuk daftar AWS sumber daya yang mendukung deteksi drift, lihat [Dukungan jenis sumber daya](#).

Selain itu, CloudFormation mendukung deteksi drift pada tipe sumber daya pribadi yang dapat disediakan; yaitu, yang tipe penyediaannya adalah salah satu atau. FULLY_MUTABLE IMMUTABLE Untuk melakukan deteksi penyimpangan pada sumber daya dari jenis sumber daya pribadi, versi default dari jenis sumber daya yang telah Anda daftarkan di akun Anda harus tersedia. Untuk informasi selengkapnya tentang jenis penyediaan sumber daya, lihat ProvisioningType parameter [DescribeType](#) tindakan di Referensi AWS CloudFormation API dan [DescribeType](#) perintah di Referensi AWS CLI Perintah. Untuk informasi lebih lanjut tentang sumber daya pribadi, lihat [Mengelola ekstensi dengan CloudFormation registri](#).

Anda dapat melakukan deteksi penyimpangan pada tumpukan dengan status berikut: CREATE_COMPLETE, UPDATE_COMPLETE, UPDATE_ROLLBACK_COMPLETE, dan UPDATE_ROLLBACK_FAILED.

Saat mendeteksi penyimpangan pada tumpukan, CloudFormation tidak mendeteksi penyimpangan pada tumpukan bersarang yang termasuk dalam tumpukan itu. Untuk informasi selengkapnya, lihat [Pisahkan template menjadi potongan-potongan yang dapat digunakan kembali menggunakan tumpukan bersarang](#). Sebaliknya, Anda dapat memulai operasi deteksi penyimpangan langsung pada tumpukan nested.

Note

CloudFormation hanya menentukan penyimpangan untuk nilai properti yang ditetapkan secara eksplisit, baik melalui template tumpukan atau dengan menentukan parameter template. Ini tidak termasuk nilai default untuk properti sumber daya. Untuk CloudFormation

melacak properti sumber daya untuk tujuan menentukan penyimpangan, tetapkan nilai properti secara eksplisit, bahkan jika Anda menyetelnya ke nilai default.

Kode status deteksi penyimpangan

Tabel di bagian ini menjelaskan berbagai jenis status yang digunakan dengan deteksi penyimpangan:

- Status operasi deteksi penyimpangan menjelaskan status operasi penyimpangan saat ini.
- Status melayang

Untuk set tumpukan, ini menjelaskan status penyimpangan set tumpukan secara keseluruhan, berdasarkan status penyimpangan dari instans tumpukan yang ada di set tersebut.

Untuk instans tumpukan, ini menjelaskan status penyimpangan dari instans tumpukan, berdasarkan status penyimpangan tumpukan terkait.

Untuk tumpukan, ini menjelaskan status penyimpangan tumpukan secara keseluruhan, berdasarkan status penyimpangan dari sumber dayanya.

- Status penyimpangan sumber daya menjelaskan status penyimpangan dari sumber daya individual.

Tabel berikut mencantumkan kode status yang ditetapkan untuk CloudFormation operasi deteksi drift tumpukan.

Status operasi deteksi penyimpangan	Deskripsi
DETECTION_COMPLETE	Operasi deteksi penyimpangan tumpukan telah berhasil diselesaikan untuk semua sumber daya di tumpukan yang mendukung deteksi penyimpangan.
DETECTION_FAILED	Operasi deteksi penyimpangan tumpukan gagal untuk setidaknya satu sumber daya dalam tumpukan. Hasil akan tersedia untuk sumber daya yang CloudFormation berhasil menyelesaikan deteksi drift.

Status operasi deteksi penyimpangan	Deskripsi
DETECTION_IN_PROGRESS	Operasi deteksi penyimpangan tumpukan saat ini sedang berlangsung.

Tabel berikut mencantumkan kode status drift yang ditetapkan ke CloudFormation tumpukan.

Status penyimpangan	Deskripsi
DRIFTED	Untuk tumpukan: Tumpukan berbeda, atau telah menyimpang, dari konfigurasi templat yang diharapkan. Tumpukan dianggap telah menyimpang jika satu atau lebih dari sumber dayanya menyimpang. Untuk instans tumpukan: Sebuah instans tumpukan dianggap menyimpang jika tumpukan yang terkait dengannya telah melayang. Untuk set tumpukan: Satu set tumpukan dianggap menyimpang jika satu atau lebih instans tumpukan telah menyimpang.
NOT_CHECKED	CloudFormation belum memeriksa apakah stack, stack set, atau stack instance berbeda dari konfigurasi template yang diharapkan.
IN_SYNC	Konfigurasi saat ini dari setiap sumber daya yang didukung cocok dengan konfigurasi templat yang diharapkan. Sebuah tumpukan, set tumpukan, atau instans tumpukan tanpa sumber daya yang mendukung deteksi penyimpangan juga akan memiliki status IN_SYNC.

Tabel berikut mencantumkan kode status drift yang ditetapkan ke CloudFormation sumber daya tumpukan.

Status drift sumber daya	Deskripsi
DELETED	Sumber daya berbeda dari konfigurasi templat yang diharapkan karena sumber daya telah dihapus.
MODIFIED	Sumber daya berbeda dari konfigurasi templat yang diharapkan.
NOT_CHECKED	CloudFormation belum memeriksa apakah sumber daya berbeda dari konfigurasi template yang diharapkan.
IN_SYNC	Konfigurasi sumber daya saat ini cocok dengan konfigurasi templat yang diharapkan.

Tabel berikut mencantumkan kode CloudFormation status tipe perbedaan yang ditetapkan ke properti sumber daya yang berbeda dari konfigurasi templat yang diharapkan.

Jenis perbedaan properti	Deskripsi
ADD	Nilai telah ditambahkan ke properti sumber daya yang merupakan tipe data array atau daftar.
REMOVE	Properti telah dihapus dari konfigurasi sumber daya saat ini.
NOT_EQUAL	Nilai properti saat ini berbeda dari nilai yang diharapkan seperti yang didefinisikan dalam templat tumpukan.

Pertimbangan saat mendeteksi penyimpangan

Agar berhasil melakukan deteksi penyimpangan pada tumpukan, pengguna harus memiliki izin berikut:

- Baca izin untuk setiap sumber daya yang mendukung deteksi penyimpangan termasuk dalam tumpukan. Misalnya, jika tumpukan mencakup sumber daya `AWS::EC2::Instance`, Anda harus memiliki izin `ec2:DescribeInstances` untuk melakukan deteksi penyimpangan pada tumpukan.
- `cloudformation:DetectStackDrift`
- `cloudformation:DetectStackResourceDrift`

- `cloudformation:BatchDescribeTypeConfigurations`

Untuk informasi selengkapnya tentang menyetel izin di CloudFormation, lihat [Kontrol CloudFormation akses dengan AWS Identity and Access Management](#).

Dalam kasus tepi tertentu, CloudFormation mungkin tidak selalu dapat mengembalikan hasil drift yang akurat. Anda harus menyadari kasus edge ini untuk benar menafsirkan hasil deteksi penyimpangan Anda.

- Dalam kasus tertentu, objek yang terdapat dalam array properti akan dilaporkan sebagai penyimpangan, padahal sebenarnya objek tersebut adalah nilai default yang diberikan ke properti dari layanan dasar yang bertanggung jawab atas sumber daya.
- Sumber daya tertentu memiliki hubungan lampiran dengan sumber daya terkait, sehingga sumber daya sebenarnya dapat melampirkan atau menghapus nilai properti untuk sumber daya lain, yang ditentukan dalam templat yang sama atau lainnya. Misalnya, sumber daya `AWS::EC2::SecurityGroupIngress` dan `AWS::EC2::SecurityGroupEgress` dapat digunakan untuk melampirkan dan menghapus nilai dari sumber daya `AWS::EC2::SecurityGroup`. Dalam kasus ini, CloudFormation analisis template tumpukan untuk lampiran sebelum melakukan perbandingan drift. Namun, tidak CloudFormation dapat melakukan analisis ini di seluruh tumpukan, sehingga mungkin tidak mengembalikan hasil penyimpangan yang akurat di mana sumber daya yang dilampirkan berada di tumpukan yang berbeda.

Sumber daya yang mendukung deteksi penyimpangan dan memungkinkan atau memerlukan lampiran dari sumber daya lain meliputi:

Jenis sumber daya	Jenis sumber daya lampiran
<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>

Jenis sumber daya	Jenis sumber daya lampiran
<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 tidak melakukan deteksi drift pada `KMSKeyId` properti sumber daya apa pun. Karena AWS KMS kunci dapat direferensikan oleh beberapa alias, tidak CloudFormation dapat menjamin hasil drift yang akurat secara konsisten untuk properti ini.
- Ada properti sumber daya tertentu yang dapat Anda tentukan dalam template tumpukan Anda yang, pada dasarnya, tidak CloudFormation akan dapat dibandingkan dengan properti di sumber daya tumpukan yang dihasilkan. Oleh karena itu, properti-properti ini tidak dapat dimasukkan dalam hasil deteksi penyimpangan. Properti semacam itu terbagi dalam dua kategori besar:
 - Nilai properti yang CloudFormation tidak dapat memetakan kembali ke nilai properti sumber daya awal mereka di template tumpukan.

Misalnya, CloudFormation tidak dapat memetakan kode sumber fungsi Lambda kembali ke jenis [Code](#) properti [AWS::Lambda::Function](#) sumber daya, dan oleh karena itu tidak CloudFormation dapat menyertakannya dalam hasil deteksi drift.

- Nilai properti yang tidak dikembalikan oleh layanan yang bertanggung jawab atas sumber daya.

Ada nilai properti tertentu yang, sesuai rancangannya, tidak pernah dikembalikan oleh layanan tempat sumber daya itu berada. Ini cenderung berisi informasi rahasia, seperti kata sandi atau data sensitif lainnya yang tidak boleh diekspos. Misalnya, layanan IAM tidak akan pernah mengembalikan nilai properti tipe Password [AWS::IAM::User LoginProfile](#) properti, dan oleh karena itu tidak CloudFormation dapat menyertakannya dalam hasil deteksi drift.

- Objek dalam array mungkin sebenarnya default layanan dan tidak drift ditambahkan secara manual.
- Jika Anda menemukan positif palsu, kirimkan komentar Anda kepada kami menggunakan tautan umpan balik di CloudFormation konsol, atau hubungi kami melalui [AWS re:Post](#).
- Beberapa properti dapat memiliki nilai input yang sama tetapi tidak identik. Untuk menghindari kesalahan positif, Anda harus memastikan bahwa konfigurasi yang Anda harapkan cocok dengan konfigurasi yang sebenarnya.

- Misalnya, konfigurasi properti sumber daya yang diharapkan dapat 1024 MB dan konfigurasi aktual dari properti sumber daya yang sama dapat 1GB. 1024 MB dan 1GB sama tetapi tidak identik.

Ketika deteksi drift berjalan pada properti sumber daya ini, itu akan menandakan hasil drifted.

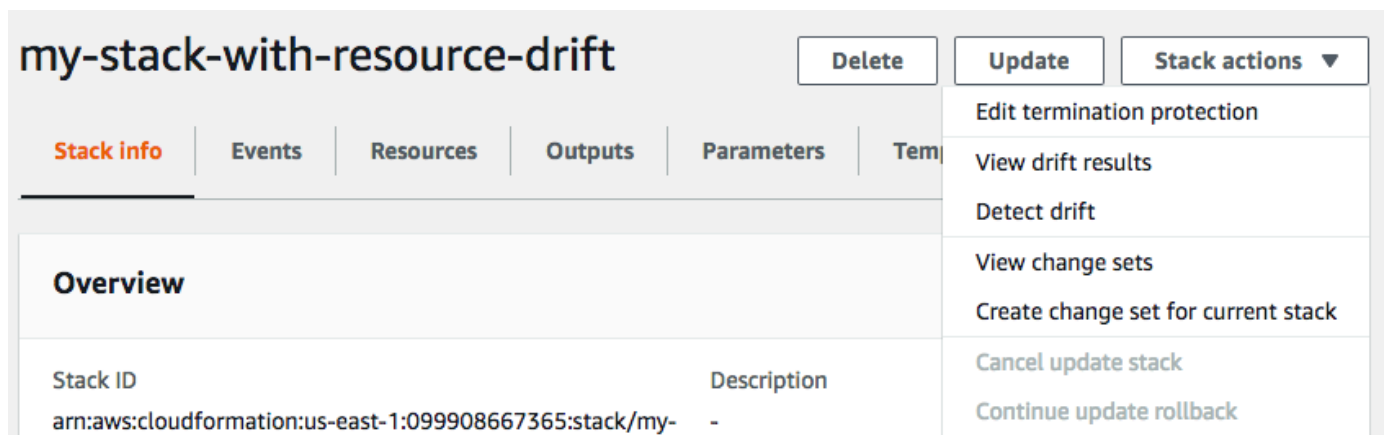
Untuk menghindari positif palsu ini, ubah konfigurasi yang diharapkan dari properti resource menjadi 1024MB dan kemudian jalankan deteksi drift.

Mendeteksi penyimpangan di seluruh tumpukan CloudFormation

Melakukan operasi deteksi penyimpangan pada tumpukan menentukan apakah tumpukan telah menyimpang dari konfigurasi templat yang diharapkan, dan mengembalikan informasi terperinci tentang status penyimpangan setiap sumber daya dalam tumpukan yang mendukung deteksi penyimpangan.

Untuk mendeteksi penyimpangan pada seluruh tumpukan menggunakan AWS Management Console

1. Buka AWS CloudFormation konsol di <https://console.aws.amazon.com/cloudformation>.
2. Dari daftar tumpukan, pilih tumpukan tempat Anda ingin melakukan deteksi penyimpangan. Di panel detail tumpukan, pilih Tindakan tumpukan, lalu pilih Deteksi penyimpangan.



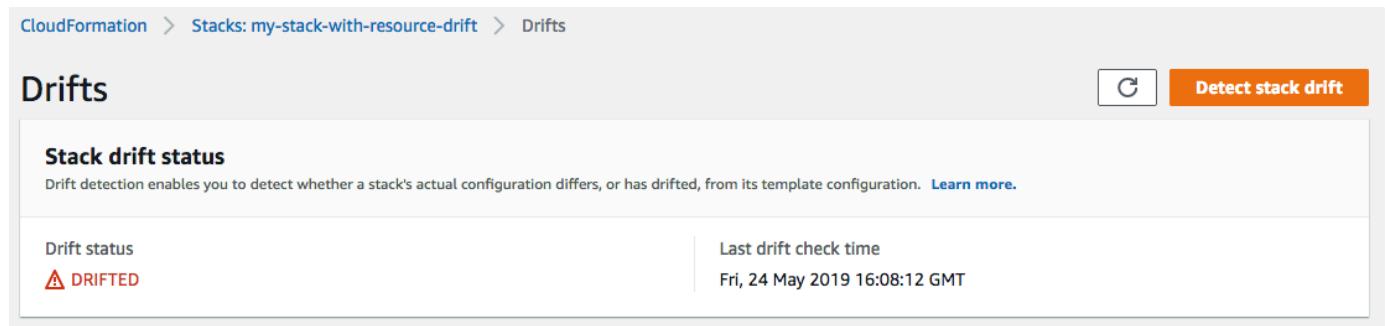
CloudFormation menampilkan bilah informasi yang menyatakan bahwa deteksi drift telah dimulai untuk tumpukan yang dipilih.

3. Tunggu hingga CloudFormation menyelesaikan operasi deteksi drift. Saat operasi deteksi drift selesai, CloudFormation perbarui status Drift dan waktu pemeriksaan drift terakhir untuk tumpukan Anda. Bidang ini tercantum di bagian Gambaran Umum di panel Info tumpukan dari halaman detail tumpukan.

Operasi deteksi penyimpangan mungkin memakan waktu beberapa menit, tergantung pada jumlah sumber daya yang disertakan dalam tumpukan. Anda hanya dapat menjalankan operasi deteksi drift tunggal pada tumpukan tertentu secara bersamaan. CloudFormation melanjutkan operasi deteksi drift bahkan setelah Anda mengabaikan bilah informasi.

4. Meninjau hasil deteksi penyimpangan untuk tumpukan dan sumber daya. Dengan tumpukan Anda dipilih, dari menu Tindakan tumpukan pilih Lihat hasil penyimpangan.

CloudFormation mencantumkan status drift keseluruhan tumpukan, selain deteksi drift terakhir kali dimulai pada tumpukan atau sumber daya individualnya. Tumpukan dianggap telah menyimpang jika satu atau lebih dari sumber dayanya menyimpang.



Di bagian Status penyimpangan sumber daya, CloudFormation daftar setiap sumber daya tumpukan, status driftnya, dan deteksi drift terakhir kali dimulai pada sumber daya. ID logis dan ID fisik setiap sumber daya ditampilkan untuk membantu Anda mengidentifikasinya. Selain itu, untuk sumber daya dengan status MODIFIED, CloudFormation menampilkan detail penyimpangan sumber daya.

Anda dapat mengurutkan sumber daya berdasarkan status penyimpangannya menggunakan kolom Status Penyimpangan.

- Untuk melihat detail sumber daya yang dimodifikasi.
 - Dengan sumber daya yang dimodifikasi dipilih, pilih Lihat detail penyimpangan.

CloudFormation menampilkan halaman detail drift untuk sumber daya itu. Halaman ini mencantumkan nilai properti yang diharapkan dan saat ini dari sumber daya, dan perbedaan apa pun di antara keduanya.

Untuk menyoroti perbedaan, di bagian Perbedaan pilih nama properti.

- Properti yang ditambahkan disorot dalam warna hijau di kolom Saat ini di bagian Detail.
- Properti yang dihapus disorot dalam warna merah di kolom Diharapkan di bagian Detail.
- Properti yang nilainya telah diubah disorot warna kuning di kolom Diharapkan dan Saat ini.

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
  }
}
```

Untuk mendeteksi penyimpangan pada seluruh tumpukan menggunakan AWS CLI

 Important

Tinjau waktu pemeriksaan drift terakhir untuk tumpukan dan konfirmasikan bahwa itu lebih awal dari stempel waktu yang ditunjukkan dalam hasil drift sumber daya untuk mencegah penggunaan data basi.

Untuk mendeteksi penyimpangan pada seluruh tumpukan menggunakan AWS CLI, gunakan AWS CLI perintah berikut:

- `detect-stack-drift` untuk memulai operasi deteksi penyimpangan di tumpukan.
 - `describe-stack-drift-detection-status` untuk memantau status operasi deteksi penyimpangan tumpukan.
 - `describe-stack-resource-drifts` untuk meninjau detail operasi deteksi penyimpangan tumpukan.
1. Gunakan `detect-stack-drift` untuk mendeteksi penyimpangan di seluruh tumpukan. Tentukan nama tumpukan atau ARN. Anda juga dapat menentukan logika IDs sumber daya tertentu yang ingin Anda gunakan sebagai filter untuk operasi deteksi drift ini.

```
aws cloudformation detect-stack-drift --stack-name my-stack-with-resource-drift
```

Output:

```
{
  "StackDriftDetectionId": "624af370-311a-11e8-b6b7-500cexample"
}
```

2. Karena operasi deteksi penyimpangan tumpukan dapat berjalan lama, gunakan `describe-stack-drift-detection-status` untuk memantau status operasi penyimpangan. Perintah ini mengambil ID deteksi penyimpangan tumpukan yang dikembalikan oleh perintah `detect-stack-drift`.

Dalam contoh di bawah ini, kami telah mengambil ID deteksi penyimpangan tumpukan yang dikembalikan oleh contoh `detect-stack-drift` di atas dan meneruskannya sebagai parameter ke `describe-stack-drift-detection-status`. Parameter mengembalikan detail operasi yang menunjukkan bahwa operasi deteksi penyimpangan telah selesai, sumber daya tumpukan

tunggal telah menyimpang, dan bahwa seluruh tumpukan dianggap telah menyimpang sebagai hasilnya.

```
aws cloudformation describe-stack-drift-detection-status --stack-drift-detection-id 624af370-311a-11e8-b6b7-500cexample
```

Output:

```
{
  "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. Ketika operasi deteksi penyimpangan tumpukan selesai, gunakan perintah `describe-stack-resource-drifts` untuk meninjau hasil, termasuk nilai properti aktual dan yang diharapkan untuk sumber daya yang telah menyimpang.

Contoh di bawah ini menggunakan `--stack-resource-drift-status-filters` opsi untuk meminta informasi stack drift untuk sumber daya yang telah dimodifikasi atau dihapus. Permintaan mengembalikan informasi tentang satu sumber daya yang telah dimodifikasi, termasuk detail tentang dua propertinya yang nilainya telah diubah. Tidak ada sumber daya yang telah dihapus.

```
aws cloudformation describe-stack-resource-drifts --stack-name my-stack-with-resource-drift --stack-resource-drift-status-filters MODIFIED DELETED
```

Output:

```
{
  "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\"
  }
]
}

```

Mendeteksi penyimpangan di sumber daya tumpukan individu

Anda dapat mendeteksi penyimpangan di sumber daya tertentu dalam satu tumpukan, bukan di seluruh tumpukan. Ini sangat berguna ketika Anda hanya perlu menentukan apakah sumber daya tertentu sekarang cocok dengan konfigurasi templat yang diharapkan lagi.

Saat melakukan deteksi drift pada sumber daya, CloudFormation juga memperbarui status tumpukan drift keseluruhan dan waktu pemeriksaan drift terakhir, jika berlaku. Misalnya, tumpukan memiliki status drift. `IN_SYNC` Anda telah CloudFormation melakukan deteksi drift pada satu atau lebih sumber daya yang terkandung dalam tumpukan itu, dan CloudFormation mendeteksi bahwa satu atau lebih sumber daya tersebut telah hanyut. CloudFormation memperbarui status stack drift ke `DRIFTED`. Sebaliknya, misalkan Anda memiliki tumpukan dengan status drift `DRIFTED` karena sumber daya hanyut tunggal. Jika Anda menyetel sumber daya itu kembali ke nilai properti yang diharapkan, dan kemudian mendeteksi penyimpangan pada sumber daya lagi, CloudFormation akan memperbarui status drift sumber daya dan status drift tumpukan `IN_SYNC` tanpa mengharuskan Anda mendeteksi penyimpangan di seluruh tumpukan lagi.

Untuk mendeteksi penyimpangan pada sumber daya individu menggunakan AWS Management Console

1. Buka AWS CloudFormation konsol di <https://console.aws.amazon.com/cloudformation>.
2. Dari daftar tumpukan, pilih tumpukan yang berisi sumber daya tersebut. CloudFormation menampilkan detail tumpukan untuk tumpukan tersebut.
3. Di panel navigasi kiri, di bawah Tumpukan, pilih Tindakan tumpukan, lalu pilih Deteksi penyimpangan.
4. Di bawah Status penyimpangan sumber daya, pilih sumber daya dan kemudian pilih Deteksi penyimpangan untuk sumber daya.

CloudFormation melakukan deteksi drift pada sumber daya yang dipilih. Jika berhasil, CloudFormation perbarui status drift sumber daya, dan status tumpukan drift keseluruhan, jika perlu. CloudFormation juga memperbarui stempel waktu kapan deteksi drift terakhir dilakukan pada sumber daya, dan tumpukan secara keseluruhan. Jika sumber daya telah dimodifikasi, CloudFormation menampilkan informasi drift terperinci tentang nilai properti yang diharapkan dan saat ini dari sumber daya.

5. Meninjau hasil deteksi penyimpangan untuk sumber daya.
 - Untuk melihat detail sumber daya yang dimodifikasi.
 - Dengan sumber daya yang dimodifikasi dipilih, pilih Lihat detail penyimpangan.

CloudFormation menampilkan detail drift untuk sumber daya tersebut, termasuk nilai properti yang diharapkan dan saat ini sumber daya, dan perbedaan apa pun di antara keduanya.

Untuk menyoroti perbedaan, di bagian Perbedaan pilih nama properti.

- Properti yang ditambahkan disorot dalam warna hijau di kolom Saat ini di bagian Detail.
- Properti yang dihapus disorot dalam warna merah di kolom Diharapkan di bagian Detail.
- Properti yang nilainya telah diubah disorot warna kuning di kolom Diharapkan dan Saat ini.

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
  }
}

```


Untuk mendeteksi penyimpangan pada sumber daya individu menggunakan AWS CLI

Important

Tinjau waktu pemeriksaan drift terakhir untuk sumber daya tumpukan dan konfirmasi bahwa itu lebih awal dari stempel waktu yang ditunjukkan dalam hasil drift sumber daya untuk mencegah penggunaan data basi.

Untuk mendeteksi penyimpangan pada sumber daya individu menggunakan AWS CLI, gunakan `detect-stack-resource-drift` perintah. Tentukan ID logis dari sumber daya dan tumpukan di dalamnya.

Contoh berikut menjalankan operasi deteksi drift pada sumber daya tumpukan tertentu, `my-drifted-resource`. Respons mengembalikan informasi yang mengonfirmasi bahwa sumber daya telah dimodifikasi, termasuk detail tentang dua propertinya yang nilainya telah diubah.

```
aws cloudformation detect-stack-resource-drift \
  --stack-name my-stack-with-resource-drift \
  --logical-resource-id my-drifted-resource
```

Output:

```
{
  "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\"
}

```

Mengatasi penyimpangan dengan operasi impor

Mungkin ada kasus di mana konfigurasi sumber daya telah menyimpang dari konfigurasi yang dimaksudkan dan Anda ingin menerima konfigurasi baru sebagai konfigurasi yang dimaksudkan. Dalam kebanyakan kasus, Anda akan mengatasi hasil penyimpangan dengan memperbarui definisi sumber daya di templat tumpukan dengan konfigurasi baru dan kemudian melakukan pembaruan tumpukan. Namun, jika konfigurasi baru memperbarui properti sumber daya yang memerlukan penggantian, sumber daya akan dibuat ulang selama pembaruan tumpukan. Jika Anda ingin mempertahankan sumber daya yang ada, Anda dapat menggunakan fitur impor sumber daya untuk memperbarui sumber daya dan menyelesaikan hasil penyimpangan tanpa menyebabkan sumber daya diganti.

Mengatasi penyimpangan untuk sumber daya melalui operasi impor terdiri dari langkah-langkah dasar berikut:

- [Tambahkan DeletionPolicy atribut, atur ke Retain, ke sumber daya](#). Hal ini memastikan sumber daya yang ada dipertahankan bukannya dihapus saat dihilangkan dari tumpukan.

- [Hapus sumber daya dari templat dan jalankan operasi pembaruan tumpukan](#). Hal ini akan menghilangkan sumber daya dari tumpukan, tetapi tidak menghapusnya.
- [Jelaskan status aktual sumber daya dalam templat tumpukan, lalu impor kembali sumber daya yang ada ke dalam tumpukan](#). Hal ini menambahkan sumber daya kembali ke tumpukan dan mengatasi perbedaan properti yang menyebabkan hasil penyimpangan.

Untuk informasi selengkapnya tentang impor sumber daya, lihat [Impor AWS sumber daya ke CloudFormation tumpukan secara manual](#). Untuk daftar sumber daya yang mendukung impor, lihat [Dukungan jenis sumber daya](#).

Dalam contoh ini, kita menggunakan template berikut, bernamatemplateToImport.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

```

Dalam contoh ini, mari kita asumsikan pengguna mengubah sumber daya di luar CloudFormation. Setelah menjalankan deteksi penyimpangan, kita menemukan bahwa GamesTable telah dimodifikasi BillingMode menjadi PAY_PER_REQUEST. Untuk informasi lebih lanjut tentang deteksi drift, lihat [Mendeteksi perubahan konfigurasi yang tidak terkelola pada tumpukan dan sumber daya dengan deteksi drift](#).

Differences (3)				
☒	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>

Tumpukan kita sekarang sudah kedaluwarsa, sumber daya kita aktif, tetapi kita ingin mempertahankan konfigurasi sumber daya yang dimaksud. Kita dapat melakukan ini dengan mengatasi penyimpangan melalui operasi impor, tanpa mengganggu layanan.

Selesaikan drift dengan operasi impor menggunakan konsol CloudFormation

Langkah 1. Memperbarui tumpukan dengan kebijakan penghapusan Pertahankan

Untuk memperbarui tumpukan menggunakan atribut **DeletionPolicy** dengan opsi **Retain**

1. Masuk ke AWS Management Console dan buka AWS CloudFormation konsol di <https://console.aws.amazon.com/cloudformation>.
2. Pada halaman Stacks, pilih tumpukan yang telah hanyut.
3. Pilih Perbarui, lalu pilih Ganti templat saat ini dari panel detail tumpukan.
4. Di halaman Tentukan templat, berikan templat terbaru Anda yang berisi atribut **DeletionPolicy** dengan opsi **Retain** menggunakan salah satu metode berikut:
 - Pilih URL Amazon S3, lalu tentukan URL untuk templat Anda di kotak teks.
 - Pilih Unggah file templat, lalu telusuri templat Anda.

Lalu, pilih Selanjutnya.

5. Tinjau halaman Tentukan detail tumpukan dan pilih Selanjutnya.
6. Tinjau halaman Konfigurasi opsi tumpukan, dan pilih Selanjutnya.
7. Pada *stack-name* halaman Review, pilih Update stack.

Hasil: Di halaman Peristiwa dari tumpukan Anda, statusnya adalah UPDATE_COMPLETE.

Untuk mengatasi drift melalui operasi impor, tanpa mengganggu layanan, tentukan sumber daya yang **Retain** [DeletionPolicy](#) ingin Anda hapus dari tumpukan Anda. Dalam contoh berikut, kita telah menambahkan [DeletionPolicy](#) atribut, set ke **Retain**, ke `GameTable` sumber daya.

Example JSON

```
"GameTable": {
  "Type": "AWS::DynamoDB::Table",
  "DeletionPolicy": "Retain",
  "Properties": {
    "TableName": "Games",
```

Example YAML

```
GameTable:
```

```
Type: 'AWS::DynamoDB::Table'
DeletionPolicy: Retain
Properties:
  TableName: Games
```

Langkah 2. Menghapus sumber daya menyimpang, parameter terkait, dan output

Untuk menghapus sumber daya menyimpang, parameter terkait, dan output

1. Pilih Perbarui, lalu pilih Ganti templat saat ini dari panel detail tumpukan.
2. Di halaman Tentukan templat, berikan templat Anda yang diperbarui dengan sumber dayanya, parameter terkait, dan output yang dihapus dari templat tumpukan menggunakan salah satu metode berikut:
 - Pilih URL Amazon S3, lalu tentukan URL untuk templat Anda di kotak teks.
 - Pilih Unggah file templat, lalu telusuri templat Anda.

Lalu, pilih Selanjutnya.

3. Tinjau halaman Tentukan detail tumpukan dan pilih Selanjutnya.
4. Tinjau halaman Konfigurasi opsi tumpukan, dan pilih Selanjutnya.
5. Pada *stack-name* halaman Review, pilih Update stack.

Hasil: ID Logis GamesTable memiliki status DELETE_SKIPPED di halaman Peristiwa dari tumpukan Anda.

Tunggu hingga CloudFormation menyelesaikan operasi pembaruan tumpukan. Setelah operasi pembaruan tumpukan selesai, hapus sumber daya, parameter terkait, dan output dari templat tumpukan. Kemudian, impor templat yang telah diperbarui. Setelah menyelesaikan tindakan ini, templat contoh sekarang terlihat seperti berikut ini.

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

```

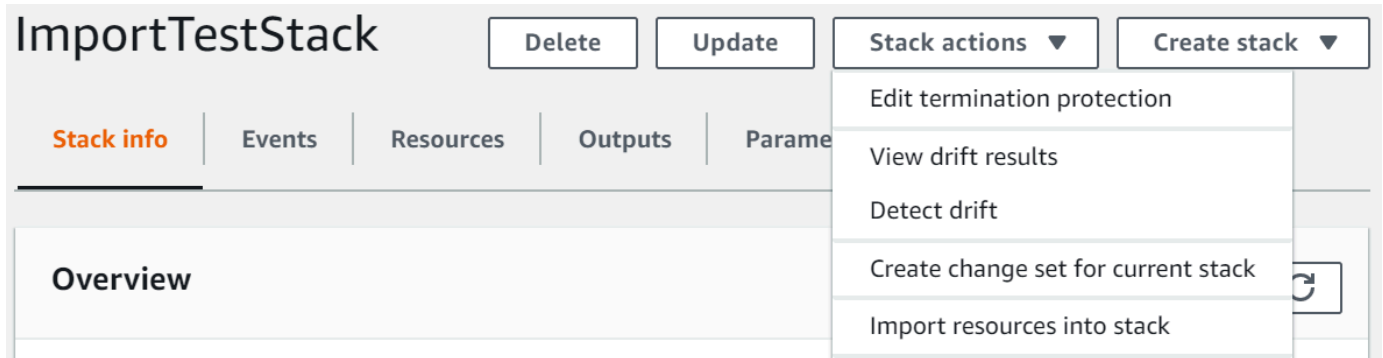
AWSTemplateFormatVersion: 2010-09-09
Description: Import test
Resources:
  ServiceTable:
    Type: 'AWS::DynamoDB::Table'
    Properties:
      TableName: Service
      AttributeDefinitions:
        - AttributeName: key
          AttributeType: S
      KeySchema:
        - AttributeName: key
          KeyType: HASH
      BillingMode: PROVISIONED
      ProvisionedThroughput:
        ReadCapacityUnits: 5
        WriteCapacityUnits: 1

```


Langkah 3. Perbarui templat untuk mencocokkan status aktif sumber daya Anda

Untuk memperbarui templat untuk mencocokkan status aktif sumber daya Anda

1. Untuk mengimpor template yang diperbarui, pilih Tindakan tumpukan lalu pilih Impor sumber daya ke tumpukan.



2. Tinjau halaman Gambaran umum impor daftar hal-hal yang harus Anda berikan selama operasi ini, lalu pilih Selanjutnya.
3. Di halaman Tentukan templat, berikan templat Anda yang telah diperbarui menggunakan salah satu metode berikut:
 - Pilih URL Amazon S3, lalu tentukan URL untuk templat Anda di kotak teks.
 - Pilih Unggah file templat, lalu telusuri templat Anda.

Lalu, pilih Selanjutnya.

4. Di halaman Identifikasi sumber daya, identifikasi setiap sumber daya target. Untuk informasi selengkapnya, lihat [Pengidentifikasi sumber daya](#).
 - a. Di bawah Properti pengenalan, pilih jenis pengenalan sumber daya. Misalnya, properti `TableName` mengidentifikasi sumber daya `AWS::DynamoDB::Table`.
 - b. Di bawah Nilai pengenalan, masukkan nilai properti yang sebenarnya. Dalam contoh templat, `TableName` untuk sumber daya `GameTable` adalah `Games`.
 - c. Pilih Berikutnya.
5. Tinjau halaman Tentukan detail tumpukan, dan pilih Selanjutnya.
6. Di halaman Gambaran umum impor, tinjau sumber daya yang diimpor, kemudian pilih Impor sumber daya. Ini akan mengimpor jenis sumber daya `AWS::DynamoDB::Table` kembali ke tumpukan Anda.

Hasil: Dalam contoh ini, kita mengatasi penyimpangan sumber daya melalui operasi impor, tanpa mengganggu layanan. Anda dapat memeriksa kemajuan tindakan impor di CloudFormation konsol di tab Acara. Sumber daya yang diimpor akan memiliki status `IMPORT_COMPLETE` diikuti oleh status `CREATE_COMPLETE` dengan Impor sumber daya selesai sebagai alasan status.

Tunggu hingga CloudFormation menyelesaikan operasi pembaruan tumpukan. Setelah operasi pembaruan tumpukan selesai, perbarui templat Anda agar sesuai dengan status aktual sumber daya Anda yang menyimpang. Misalnya, `BillingMode` akan diatur ke `PAY_PER_REQUEST` dan `ReadCapacityUnits` serta `WriteCapacityUnits` akan diatur ke `0`.

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

```

```
GameTable:
  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
```

Impor AWS sumber daya ke CloudFormation tumpukan

Anda dapat mengimpor sumber daya yang ada ke dalam CloudFormation tumpukan. Ini berguna jika Anda ingin mulai menggunakan CloudFormation untuk mengelola sumber daya yang dibuat di luar CloudFormation, tanpa harus menghapus dan membuatnya kembali.

CloudFormation menawarkan opsi berikut untuk mengimpor sumber daya yang ada ke tumpukan:

- [Generator IAC](#) adalah alat yang secara otomatis memindai sumber daya Anda yang ada dan menghasilkan CloudFormation template berdasarkan statusnya saat ini. Template ini kemudian dapat digunakan untuk mengimpor sumber daya tersebut ke dalam tumpukan.
- [Impor sumber daya](#) adalah proses manual di mana Anda menjelaskan sumber daya yang ada di CloudFormation template Anda dan kemudian mengimpornya ke tumpukan. Pendekatan ini mengharuskan Anda untuk secara manual menentukan properti sumber daya dan konfigurasi dalam template.
- [Impor otomatis](#) adalah proses otomatis di mana Anda mendeskripsikan sumber daya yang ada di CloudFormation template Anda dan CloudFormation mengimpor yang dengan nama kustom yang cocok ke dalam tumpukan.
- [Stack refactoring](#) adalah fitur yang menyederhanakan penataan ulang sumber daya di CloudFormation tumpukan Anda sambil tetap mempertahankan properti dan data sumber daya yang ada. Dengan stack refactoring, Anda dapat memindahkan sumber daya antar tumpukan,

membagi tumpukan monolitik menjadi komponen yang lebih kecil, atau mengkonsolidasikan beberapa tumpukan menjadi satu.

Selain CloudFormation mengelola sumber daya yang ada, fitur impor sumber daya dapat berguna dalam skenario berikut:

- Memindahkan sumber daya antar tumpukan — Anda dapat mengimpor sumber daya dari satu tumpukan ke tumpukan lainnya, memungkinkan Anda mengatur ulang infrastruktur sesuai kebutuhan.
- Menyusun tumpukan yang ada — Anda dapat mengimpor tumpukan yang ada sebagai tumpukan bersarang di dalam tumpukan lain, memungkinkan desain infrastruktur modular dan dapat digunakan kembali.

CloudFormation mendukung mengimpor berbagai sumber daya. Lihat informasi yang lebih lengkap di [Dukungan jenis sumber daya](#).

Topik

- [Impor AWS sumber daya ke CloudFormation tumpukan secara manual](#)
- [Impor AWS sumber daya ke CloudFormation tumpukan secara otomatis](#)
- [Pengembalian operasi impor](#)

Impor AWS sumber daya ke CloudFormation tumpukan secara manual

Dengan impor sumber daya, Anda dapat mengimpor AWS sumber daya yang ada ke CloudFormation tumpukan baru atau yang sudah ada. Selama operasi impor, Anda membuat perubahan set yang mengimpor sumber daya yang ada ke tumpukan atau membuat tumpukan baru dari sumber daya yang ada. Anda memberikan hal berikut selama impor.

- Templat yang menggambarkan seluruh tumpukan, termasuk sumber daya tumpukan asli dan sumber daya yang Anda impor. Setiap sumber daya yang akan diimpor harus memiliki [DeletionPolicy](#) atribut.
- Pengidentifikasi untuk sumber daya yang Anda impor yang CloudFormation dapat digunakan untuk memetakan logika IDs dalam template dengan sumber daya yang ada.

 Note

CloudFormation hanya mendukung satu tingkat bersarang menggunakan impor sumber daya. Ini berarti Anda tidak dapat mengimpor tumpukan ke tumpukan anak atau mengimpor tumpukan yang memiliki anak.

Topik

- [Pengidentifikasi sumber daya](#)
- [Validasi impor sumber daya](#)
- [Kode status impor sumber daya](#)
- [Pertimbangan selama operasi impor](#)
- [Sumber daya tambahan](#)
- [Membuat tumpukan dari sumber daya yang ada](#)
- [Mengimpor sumber daya yang ada ke dalam tumpukan](#)
- [Memindahkan sumber daya antar tumpukan](#)
- [Membuat nest tumpukan yang ada](#)

Pengidentifikasi sumber daya

Anda memberikan dua nilai untuk mengidentifikasi setiap sumber daya yang Anda impor.

- Properti pengenalan. Ini adalah properti sumber daya yang dapat digunakan untuk mengidentifikasi setiap jenis sumber daya. Misalnya, `AWS::S3::Bucket` sumber daya dapat diidentifikasi menggunakan `BucketName`.

Properti sumber daya yang Anda gunakan untuk mengidentifikasi sumber daya yang Anda impor bervariasi sesuai dengan jenis sumber daya. Anda dapat menemukan properti sumber daya di CloudFormation konsol. Setelah membuat templat yang menyertakan sumber daya yang akan diimpor, Anda dapat memulai proses impor, di mana Anda akan menemukan properti pengenalan untuk sumber daya yang Anda impor. Untuk beberapa jenis sumber daya, mungkin ada beberapa cara untuk mengidentifikasinya, dan Anda dapat memilih properti mana yang akan digunakan dalam daftar drop-down.

Atau, Anda bisa mendapatkan properti pengenalan untuk sumber daya yang Anda impor dengan memanggil perintah [get-template-summary](#) CLI dan menentukan URL S3 dari template tumpukan sebagai nilai untuk opsi. `--template-url`

- Nilai pengenalan. Ini adalah nilai properti aktual sumber daya. Misalnya, nilai aktual untuk BucketName properti mungkin MyS3Bucket.

Anda bisa mendapatkan nilai properti identifier dari konsol layanan untuk sumber daya.

Validasi impor sumber daya

Selama operasi impor, CloudFormation melakukan validasi berikut.

- Terdapat sumber daya untuk diimpor.
- Nilai properti dan konfigurasi untuk setiap sumber daya yang diimpor akan mematuhi skema jenis sumber daya, yang mendefinisikan properti yang diterima, properti yang diperlukan, dan nilai-nilai properti yang didukung.
- Properti yang dibutuhkan ditentukan dalam templat. Properti yang diperlukan untuk setiap jenis sumber daya dijelaskan dalam [referensi tipe AWS sumber daya dan properti](#).
- Sumber daya yang akan diimpor bukan milik tumpukan lain di Wilayah yang sama.

CloudFormation tidak memeriksa apakah konfigurasi template cocok dengan konfigurasi properti sumber daya yang sebenarnya.

Important

Verifikasi bahwa sumber daya dan propertinya yang ditentukan dalam templat cocok dengan konfigurasi impor sumber daya yang dimaksudkan untuk menghindari perubahan yang tidak terduga.

Kode status impor sumber daya

Tabel ini menjelaskan berbagai jenis status yang digunakan dengan fitur impor sumber daya.

Impor status operasi	Deskripsi
IMPORT_IN_PROGRESS	Operasi impor sedang berlangsung.
IMPORT_COMPLETE	Operasi impor selesai untuk semua sumber daya di tumpukan.
IMPORT_ROLLBACK_IN_PROGRESS	Operasi impor rollback memutar kembali konfigurasi template sebelumnya.
IMPORT_ROLLBACK_FAILED	Operasi rollback impor gagal.
IMPORT_ROLLBACK_COMPLETE	Impor digulung kembali ke konfigurasi template sebelumnya.

Pertimbangan selama operasi impor

- Setelah impor selesai dan sebelum melakukan operasi tumpukan berikutnya, kami sarankan untuk menjalankan deteksi drift pada sumber daya impor. Deteksi drift memastikan bahwa konfigurasi templat cocok dengan konfigurasi yang sebenarnya. Untuk informasi selengkapnya, lihat [Mendeteksi penyimpangan di seluruh tumpukan CloudFormation](#).
- Operasi impor tidak mengizinkan pembuata sumber daya baru, penghapusan sumber daya, atau perubahan pada konfigurasi properti.
- Setiap sumber daya yang diimpor harus memiliki DeletionPolicy atribut untuk operasi impor supaya berhasil. Parameter DeletionPolicy dapat diatur untuk setiap nilai yang memungkinkan. Hanya sumber daya yang Anda impor yang membutuhkan aDeletionPolicy. Sumber daya yang sudah menjadi bagian dari tumpukan tidak memerlukan DeletionPolicy.
- Anda tidak dapat mengimpor sumber daya yang sama ke beberapa tumpukan.
- Anda dapat menggunakan kondisi kebijakan cloudformation:ImportResourceTypes IAM untuk mengontrol jenis sumber daya yang dapat digunakan pengguna selama operasi impor. Untuk informasi selengkapnya, lihat [Kunci kondisi kebijakan untuk CloudFormation](#).
- Batas CloudFormation tumpukan berlaku saat mengimpor sumber daya. Untuk informasi lebih lanjut tentang batasan, lihat [Memahami CloudFormation kuota](#).

Sumber daya tambahan

Untuk mengatasi tumpukan drift dengan impor sumber daya, lihat [Mengatasi penyimpangan dengan operasi impor](#).

Membuat tumpukan dari sumber daya yang ada

Topik ini menunjukkan cara membuat tumpukan dari AWS sumber daya yang ada dengan mendeskripsikannya dalam templat. Untuk memindai sumber daya yang ada dan secara otomatis membuat templat yang dapat Anda gunakan untuk mengimpor sumber daya yang ada ke CloudFormation atau mereplikasi sumber daya di akun baru, lihat [Hasilkan templat dari sumber daya yang ada dengan generator IAc](#).

Prasyarat

Sebelum memulai, Anda harus memiliki hal-hal berikut:

- Template yang menjelaskan semua sumber daya yang Anda inginkan di tumpukan baru. Simpan template secara lokal atau di bucket Amazon S3.
- Untuk setiap sumber daya yang ingin Anda impor, sertakan yang berikut ini:
 - properti dan nilai properti yang menentukan konfigurasi sumber daya saat ini.
 - pengidentifikasi unik untuk sumber daya, seperti nama sumber daya. Untuk informasi selengkapnya, lihat [Pengidentifikasi sumber daya](#).
 - [DeletionPolicyatribut](#).

Topik

- [Contoh Templat](#)
- [Buat tumpukan dari sumber daya yang ada menggunakan AWS Management Console](#)
- [Buat tumpukan dari sumber daya yang ada menggunakan AWS CLI](#)

Contoh Templat

Dalam panduan ini, kami menganggap Anda menggunakan contoh template berikut, yang disebut `TemplateToImport.json`, yang menentukan dua tabel DynamoDB yang dibuat di luar. CloudFormation `ServiceTable` dan `GameTable` merupakan target impor.

 Note

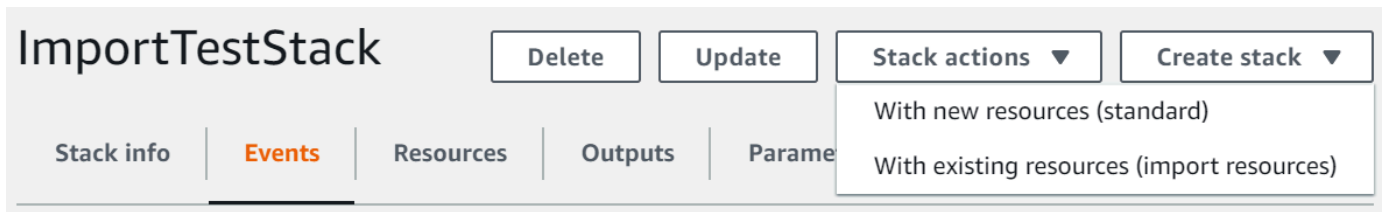
Template ini dimaksudkan sebagai contoh saja. Untuk menggunakannya untuk tujuan pengujian Anda sendiri, ganti sumber daya sampel dengan sumber daya dari akun Anda.

```
{
  "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
  }
}
}
```

Buat tumpukan dari sumber daya yang ada menggunakan AWS Management Console

1. Masuk ke AWS Management Console dan buka AWS CloudFormation konsol di <https://console.aws.amazon.com/cloudformation>.
2. Pada halaman Tumpukan, pilih Buat Tumpukan, lalu pilih Dengan sumber daya yang ada (sumber daya impor).



3. Baca halaman Gambaran Umum Impor untuk daftar hal yang harus Anda berikan selama operasi ini. Lalu, pilih Selanjutnya.
4. Di halaman Tentukan templat, berikan templat Anda menggunakan salah satu metode berikut, lalu pilih Selanjutnya.
 - Pilih URL Amazon S3, lalu tentukan URL untuk templat Anda di kotak teks.
 - Pilih Unggah file templat, lalu telusuri templat Anda.
5. Di halaman Identifikasi sumber daya, identifikasi setiap sumber daya target. Untuk informasi selengkapnya, lihat [Pengidentifikasi sumber daya](#).

- a. Di bawah Properti pengidentifikasi, pilih jenis pengidentifikasi sumber daya. Misalnya, sumber daya AWS::DynamoDB::Table dapat diidentifikasi menggunakan TableName properti.
 - b. Di bawah Nilai pengidentifikasi, ketikkan nilai properti yang sebenarnya. Misalnya, TableName untuk GamesTable sumber daya di templat contoh adalah *Games*.
 - c. Pilih Berikutnya.
6. Di halaman Tentukan detail tumpukan, ubah parameter apa pun, lalu pilih Selanjutnya. Ini secara otomatis membuat set perubahan.

Important

Operasi impor gagal jika Anda mengubah parameter yang sudah ada yang memulai operasi membuat, memperbarui, atau menghapus.

7. Pada *stack-name* halaman Tinjauan, konfirmasi bahwa sumber daya yang benar sedang diimpor, lalu pilih Impor sumber daya. Ini secara otomatis mengeksekusi perubahan set yang dibuat pada langkah terakhir.

Panel Peristiwa halaman Detail tumpukan untuk tampilan tumpukan baru Anda.

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	SamplelamUser	UPDATE_COMPLETE	-
2019-10-24 15:52:47 UTC-0700	SamplelamUser	UPDATE_IN_PROGRESS	Apply stack-level tags to imported resource if applicable.
2019-10-24 15:52:46 UTC-0700	SamplelamUser	IMPORT_COMPLETE	Resource import completed.
2019-10-24 15:52:45 UTC-0700	SamplelamUser	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. (Opsional) Jalankan deteksi drift pada tumpukan untuk memastikan templat dan konfigurasi sebenarnya cocok dengan sumber daya yang diimpor. Untuk informasi lebih lanjut tentang mendeteksi penyimpangan, lihat [Mendeteksi penyimpangan di seluruh tumpukan CloudFormation](#)

9. (Opsional) Jika sumber daya yang diimpor tidak cocok dengan konfigurasi templat yang diharapkan, perbaiki konfigurasi templat atau perbarui sumber daya secara langsung. Dalam panduan ini, kita memperbaiki konfigurasi templat untuk mencocokkan konfigurasi yang sebenarnya.
 - a. [Kembalikan operasi impor](#) untuk sumber daya yang terpengaruh.
 - b. Tambahkan target impor ke templat Anda lagi, memastikan bahwa konfigurasi templat cocok dengan konfigurasi yang sebenarnya.
 - c. Ulangi langkah 2 - 8 menggunakan template yang dimodifikasi untuk mengimpor sumber daya lagi.

Buat tumpukan dari sumber daya yang ada menggunakan AWS CLI

1. Untuk mempelajari properti mana yang mengidentifikasi setiap jenis sumber daya dalam template, jalankan `get-template-summary` perintah, tentukan URL S3 template. Misalnya, sumber daya `AWS::DynamoDB::Table` dapat diidentifikasi menggunakan `TableName` properti. Untuk `GameTable` sumber daya dalam templat contoh, nilai `TableName` adalah `Games`. Anda akan membutuhkan informasi ini di langkah berikutnya.

```
aws cloudformation get-template-summary \  
  --template-url https://amzn-s3-demo-bucket.s3.us-west-2.amazonaws.com/  
TemplateToImport.json
```

Untuk informasi selengkapnya, lihat [Pengidentifikasi sumber daya](#).

2. Buat daftar sumber daya aktual dari template Anda dan pengenalan uniknya dalam format string JSON berikut.

```
[{"ResourceType":"AWS::DynamoDB::Table","LogicalResourceId":"GameTable","ResourceIdentifier":{"TableName":"Games"}},  
{"ResourceType":"AWS::DynamoDB::Table","LogicalResourceId":"ServiceTable","ResourceIdentifier":{"TableName":"Service"}}]
```

Atau, Anda dapat menentukan parameter berformat JSON dalam file konfigurasi.

Misalnya, untuk mengimpor `ServiceTable` dan `GameTable`, Anda dapat membuat *ResourcesToImport.txt* file yang berisi konfigurasi berikut.

```
[
```

```
[
  {
    "ResourceType": "AWS::DynamoDB::Table",
    "LogicalResourceId": "GamesTable",
    "ResourceIdentifier": {
      "TableName": "Games"
    }
  },
  {
    "ResourceType": "AWS::DynamoDB::Table",
    "LogicalResourceId": "ServiceTable",
    "ResourceIdentifier": {
      "TableName": "Service"
    }
  }
]
```

3. Untuk membuat set perubahan, gunakan create-change-set perintah berikut dan ganti teks placeholder. Untuk --change-set-type opsi, tentukan nilai **IMPORT**. Untuk --resources-to-import opsi, ganti string JSON sampel dengan string JSON aktual yang baru saja Anda buat.

```
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"}}, {"ResourceType": "AWS::DynamoDB::Table", "LogicalResourceId": "ServiceTable", "ResourceIdentifier": {"TableName": "Service"}}]'
```

Note

--resources-to-import tidak mendukung YAMAL sebaris. Persyaratan untuk menghindari tanda kutip dalam string JSON bervariasi tergantung pada terminal Anda. Untuk informasi selengkapnya, lihat [Menggunakan tanda kutip di dalam string](#) di AWS Command Line Interface Panduan Pengguna.

Atau, Anda dapat menggunakan URL file sebagai masukan untuk `--resources-to-import` opsi, seperti yang ditunjukkan pada contoh berikut.

```
--resources-to-import file:///ResourcesToImport.txt
```

4. Tinjau perubahan yang ditetapkan untuk memastikan sumber daya yang benar akan diimpor.

```
aws cloudformation describe-change-set \  
  --change-set-name ImportChangeSet --stack-name TargetStack
```

5. Untuk memulai set perubahan dan mengimpor sumber daya, gunakan `execute-change-set` perintah berikut dan ganti teks placeholder. Setelah berhasil menyelesaikan operasi (`IMPORT_COMPLETE`), sumber daya berhasil diimpor.

```
aws cloudformation execute-change-set \  
  --change-set-name ImportChangeSet --stack-name TargetStack
```

6. (Opsional) Jalankan deteksi drift pada `IMPORT_COMPLETE` tumpukan untuk memastikan templat dan konfigurasi sebenarnya cocok dengan sumber daya yang diimpor. Untuk informasi lebih lanjut tentang mendeteksi penyimpangan, lihat [Mendeteksi penyimpangan di sumber daya tumpukan individu](#)
 - a. Jalankan deteksi drift pada tumpukan yang ditentukan.

```
aws cloudformation detect-stack-drift --stack-name TargetStack
```

Jika berhasil, perintah ini mengembalikan output sampel berikut.

```
{ "Stack-Drift-Detection-Id" : "624af370-311a-11e8-b6b7-500cexample" }
```

- b. Lihat kemajuan operasi deteksi drift untuk ID deteksi drift tumpukan yang ditentukan.

```
aws cloudformation describe-stack-drift-detection-status \  
  --stack-drift-detection-id 624af370-311a-11e8-b6b7-500cexample
```

- c. Lihat informasi drift untuk sumber daya yang telah diperiksa untuk drift di tumpukan yang ditentukan.

```
aws cloudformation describe-stack-resource-drifts --stack-name TargetStack
```

7. (Opsional) Jika sumber daya yang diimpor tidak cocok dengan konfigurasi templat yang diharapkan, perbaiki konfigurasi templat atau perbarui sumber daya secara langsung. Dalam panduan ini, kita memperbaiki konfigurasi templat untuk mencocokkan konfigurasi yang sebenarnya.
 - a. [Kembalikan operasi impor](#) untuk sumber daya yang terpengaruh.
 - b. Tambahkan target impor ke templat Anda lagi, memastikan bahwa konfigurasi templat cocok dengan konfigurasi yang sebenarnya.
 - c. Ulangi langkah 3 - 6 menggunakan template yang dimodifikasi untuk mengimpor sumber daya lagi.

Mengimpor sumber daya yang ada ke dalam tumpukan

Topik ini menunjukkan cara mengimpor AWS sumber daya yang ada ke tumpukan yang ada dengan mendeskripsikannya dalam templat. Untuk memindai sumber daya yang ada dan secara otomatis membuat templat yang dapat Anda gunakan untuk mengimpor sumber daya yang ada ke CloudFormation atau mereplikasi sumber daya di akun baru, lihat [Hasilkan templat dari sumber daya yang ada dengan generator IAC](#).

Prasyarat

Sebelum memulai, Anda harus memiliki hal-hal berikut:

- Templat yang menggambarkan seluruh tumpukan, termasuk kedua sumber daya yang sudah menjadi bagian dari tumpukan dan sumber daya untuk mengimpor. Simpan template secara lokal atau di bucket Amazon S3.

Untuk mendapatkan salinan template tumpukan yang sedang berjalan

1. Buka CloudFormation konsol di <https://console.aws.amazon.com/cloudformation/>.
 2. Dari daftar tumpukan, pilih tumpukan tempat Anda ingin mengambil template dari.
 3. Di panel detail tumpukan, pilih Template tab, lalu pilih Salin ke clipboard.
 4. Tempelkan kode ke editor teks untuk mulai menambahkan sumber daya lain ke template.
- Untuk setiap sumber daya yang ingin Anda impor, sertakan yang berikut ini:
 - properti dan nilai properti yang menentukan konfigurasi sumber daya saat ini.

- pengenal unik untuk sumber daya, seperti nama sumber daya. Untuk informasi selengkapnya, lihat [Pengidentifikasi sumber daya](#).
- [DeletionPolicyatribut](#).

Topik

- [Contoh Templat](#)
- [Impor sumber daya yang ada ke tumpukan menggunakan AWS Management Console](#)
- [Impor sumber daya yang ada ke tumpukan menggunakan AWS CLI](#)

Contoh Templat

Dalam panduan ini, kami menganggap Anda menggunakan contoh template berikut, yang disebut `TemplateToImport.json`, yang menentukan dua tabel DynamoDB. `ServiceTable` saat ini merupakan bagian dari tumpukan, dan `GameTable` merupakan tabel yang ingin Anda impor.

Note

Template ini dimaksudkan sebagai contoh saja. Untuk menggunakannya untuk tujuan pengujian Anda sendiri, ganti sumber daya sampel dengan sumber daya dari akun Anda.

```
{
  "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
      }
    }
  }
}

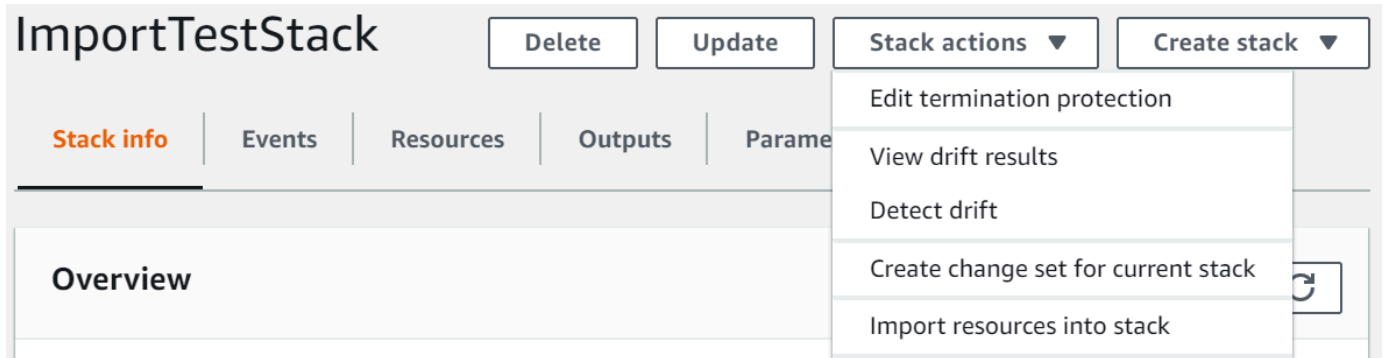
```

Impor sumber daya yang ada ke tumpukan menggunakan AWS Management Console

Note

CloudFormation Konsol tidak mendukung penggunaan fungsi intrinsik `Fn::Transform` saat mengimpor sumber daya. Anda dapat menggunakan AWS CLI untuk mengimpor sumber daya yang menggunakan `Fn::Transform` fungsi tersebut.

1. Masuk ke AWS Management Console dan buka AWS CloudFormation konsol di <https://console.aws.amazon.com/cloudformation>.
2. Pada halaman Tumpukan, pilih tumpukan yang sumber dayanya ingin Anda impor.
3. Pilih Tumpukan tindakan, lalu pilih Impor sumber daya ke tumpukan.



4. Tinjau halaman Gambaran umum impor, dan kemudian pilih Selanjutnya.
5. Di halaman Tentukan templat, berikan templat Anda yang telah diperbarui menggunakan salah satu metode berikut, lalu pilih Selanjutnya.
 - Pilih URL Amazon S3, lalu tentukan URL untuk templat Anda di kotak teks.
 - Pilih Unggah file templat, lalu telusuri templat Anda.
6. Di halaman Identifikasi sumber daya, identifikasi setiap sumber daya target. Untuk informasi selengkapnya, lihat [Pengidentifikasi sumber daya](#).
 - a. Di bawah Properti pengidentifikasi, pilih jenis pengidentifikasi sumber daya. Misalnya, sumber daya AWS::DynamoDB::Table dapat diidentifikasi menggunakan TableName properti.
 - b. Di bawah Nilai pengidentifikasi, ketikkan nilai properti yang sebenarnya. Misalnya, TableName untuk GamesTable sumber daya di templat contoh adalah **Games**.
 - c. Pilih Berikutnya.
7. Pada halaman Tentukan detail tumpukan, perbarui parameter apapun, dan kemudian pilih Selanjutnya. Ini secara otomatis membuat set perubahan.

Note

Operasi impor gagal jika Anda mengubah parameter yang sudah ada yang memulai operasi membuat, memperbarui, atau menghapus.

8. Pada **stack-name** halaman Tinjauan, tinjau sumber daya yang akan diimpor, lalu pilih Impor sumber daya. Ini secara otomatis menjalankan set perubahan yang dibuat pada langkah terakhir. Semua tanda stack-level diterapkan pada sumber daya yang diimpor saat ini. Untuk informasi selengkapnya, lihat [Konfigurasi opsi tumpukan](#).

Halaman Peristiwa untuk menampilkan tumpukan.

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

9. (Opsional) Jalankan deteksi drift pada tumpukan untuk memastikan templat dan konfigurasi sebenarnya cocok dengan sumber daya yang diimpor. Untuk informasi lebih lanjut tentang mendeteksi penyimpangan, lihat. [Mendeteksi penyimpangan di seluruh tumpukan CloudFormation](#)
10. (Opsional) Jika sumber daya yang diimpor tidak cocok dengan konfigurasi templat yang diharapkan, perbaiki konfigurasi templat atau perbarui sumber daya secara langsung. Untuk informasi selengkapnya tentang mengimpor sumber daya hanyut, lihat. [Mengatasi penyimpangan dengan operasi impor](#)

Impor sumber daya yang ada ke tumpukan menggunakan AWS CLI

1. Untuk mempelajari properti mana yang mengidentifikasi setiap jenis sumber daya dalam template, jalankan get-template-summary perintah, tentukan URL S3 template. Misalnya, sumber daya AWS::DynamoDB::Table dapat diidentifikasi menggunakan TableName properti. Untuk GamesTable sumber daya dalam templat contoh, nilai TableName adalah Games. Anda akan membutuhkan informasi ini di langkah berikutnya.

```
aws cloudformation get-template-summary \
```

```
--template-url https://amzn-s3-demo-bucket.s3.us-west-2.amazonaws.com/
TemplateToImport.json
```

Untuk informasi selengkapnya, lihat [Pengidentifikasi sumber daya](#).

2. Buat daftar sumber daya aktual untuk diimpor dan pengenalan uniknya dalam format string JSON berikut.

```
[{"ResourceType":"AWS::DynamoDB::Table","LogicalResourceId":"GamesTable","ResourceIdentifier":{"TableName":"Games"}}]
```

Atau, Anda dapat menentukan parameter berformat JSON dalam file konfigurasi.

Misalnya, untuk mengimpor `GamesTable`, Anda dapat membuat *ResourcesToImport.txt* file yang berisi konfigurasi berikut.

```
[
  {
    "ResourceType":"AWS::DynamoDB::Table",
    "LogicalResourceId":"GamesTable",
    "ResourceIdentifier": {
      "TableName":"Games"
    }
  }
]
```

3. Untuk membuat set perubahan, gunakan `create-change-set` perintah berikut dan ganti teks placeholder. Untuk `--change-set-type` opsi, tentukan nilai **IMPORT**. Untuk `--resources-to-import` opsi, ganti string JSON sampel dengan string JSON aktual yang baru saja Anda buat.

```
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 tidak mendukung YAMAL sebaris. Persyaratan untuk menghindari tanda kutip dalam string JSON bervariasi tergantung pada terminal Anda. Untuk informasi selengkapnya, lihat [Menggunakan tanda kutip di dalam string](#) di AWS Command Line Interface Panduan Pengguna.

Atau, Anda dapat menggunakan URL file sebagai masukan untuk --resources-to-import opsi, seperti yang ditunjukkan pada contoh berikut.

```
--resources-to-import file:///ResourcesToImport.txt
```

4. Tinjau perubahan yang ditetapkan untuk memastikan sumber daya yang benar akan diimpor.

```
aws cloudformation describe-change-set \  
  --change-set-name ImportChangeSet --stack-name TargetStack
```

5. Untuk memulai set perubahan dan mengimpor sumber daya, gunakan execute-change-set perintah berikut dan ganti teks placeholder. Semua tanda stack-level diterapkan pada sumber daya yang diimpor saat ini. Untuk informasi selengkapnya, lihat [Konfigurasi opsi tumpukan](#). Setelah berhasil menyelesaikan operasi (IMPORT_COMPLETE), sumber daya berhasil diimpor.

```
aws cloudformation execute-change-set \  
  --change-set-name ImportChangeSet --stack-name TargetStack
```

6. (Opsional) Jalankan deteksi drift pada IMPORT_COMPLETE tumpukan untuk memastikan templat dan konfigurasi sebenarnya cocok dengan sumber daya yang diimpor. Untuk informasi lebih lanjut tentang mendeteksi penyimpangan, lihat [Mendeteksi penyimpangan di seluruh tumpukan CloudFormation](#)

- a. Jalankan deteksi drift pada tumpukan yang ditentukan.

```
aws cloudformation detect-stack-drift --stack-name TargetStack
```

Jika berhasil, perintah ini mengembalikan output sampel berikut.

```
{ "Stack-Drift-Detection-Id" : "624af370-311a-11e8-b6b7-500cexample" }
```

- b. Lihat kemajuan operasi deteksi drift untuk ID deteksi drift tumpukan yang ditentukan.

```
aws cloudformation describe-stack-drift-detection-status \  
  --stack-drift-detection-id 624af370-311a-11e8-b6b7-500cexample
```

- c. Lihat informasi drift untuk sumber daya yang telah diperiksa untuk drift di tumpukan yang ditentukan.

```
aws cloudformation describe-stack-resource-drifts --stack-name TargetStack
```

7. (Opsional) Jika sumber daya yang diimpor tidak cocok dengan konfigurasi templat yang diharapkan, perbaiki konfigurasi templat atau perbarui sumber daya secara langsung. Untuk informasi selengkapnya tentang mengimpor sumber daya hanyut, lihat. [Mengatasi penyimpangan dengan operasi impor](#)

Memindahkan sumber daya antar tumpukan

Dengan menggunakan fitur `resource import`, Anda dapat memindahkan sumber daya antar, atau memfaktorkan ulang (refactor), tumpukan. Anda harus terlebih dahulu menambahkan kebijakan `Retain` penghapusan ke sumber daya yang ingin Anda pindahkan untuk memastikan bahwa sumber daya dipertahankan saat Anda menghapusnya dari tumpukan sumber dan mengimpornya ke tumpukan target.

Jika Anda baru mengimpor, kami sarankan Anda terlebih dahulu meninjau informasi pengantar dalam topik. [Impor AWS sumber daya ke CloudFormation tumpukan](#)

Important

Tidak semua sumber daya mendukung operasi impor. Lihat [Sumber daya yang mendukung operasi impor](#) sebelum menghapus sumber daya dari tumpukan Anda. Jika Anda menghapus sumber daya yang tidak mendukung operasi impor dari tumpukan, Anda tidak dapat mengimpor sumber daya ke tumpukan lain atau mengembalikannya ke tumpukan sumber.

Memfaktorkan ulang tumpukan menggunakan AWS Management Console

1. Dalam template sumber, tentukan sumber daya `Retain` [DeletionPolicy](#) yang ingin Anda pindahkan.

Dalam contoh sumber templat berikut, Games adalah target faktor ulang (refactor) ini.

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. Buka CloudFormation konsol untuk melakukan pembaruan tumpukan guna menerapkan kebijakan penghapusan.
 - a. Di halaman Tumpukan, dengan tumpukan yang dipilih, pilih Perbarui.
 - b. Di bawah Siapkan templat, Pilih Ganti templat saat ini.
 - c. Di bawah Tentukan templat, sediakan templat sumber yang diperbarui dengan atribut `DeletionPolicy` pada `GameTable`, lalu pilih Selanjutnya.
 - Pilih URL Amazon S3, lalu tentukan URL ke templat sumber yang diperbarui di kotak teks.
 - Pilih Unggah file templat, lalu telusuri file templat sumber yang diperbarui.
 - d. Pada halaman Tentukan detail tumpukan, tidak ada perubahan yang diperlukan. Pilih Berikutnya.
 - e. Pada halaman Konfigurasi pilihan tumpukan, tidak ada perubahan yang diperlukan. Pilih Berikutnya.
 - f. Pada *SourceStackName* halaman Ulasan, tinjau perubahan Anda. Jika template Anda berisi sumber daya IAM, pilih Saya mengakui bahwa template ini dapat membuat sumber daya IAM untuk menentukan bahwa Anda ingin menggunakan sumber daya IAM dalam template. Untuk informasi selengkapnya tentang penggunaan sumber daya IAM dalam templat, lihat [Kontrol CloudFormation akses dengan AWS Identity and Access Management](#). Lalu, perbarui tumpukan sumber Anda dengan membuat set perubahan atau perbarui tumpukan sumber Anda secara langsung.
3. Hapus sumber daya, parameter terkait, dan output dari templat sumber, lalu tambahkan ke templat target.

Templat sumber sekarang terlihat seperti berikut.

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
        }
      }
    }
  }
}
```

Contoh templat target berikut saat ini memiliki sumber daya `PlayersTable`, dan sekarang juga berisi `GamesTable`.

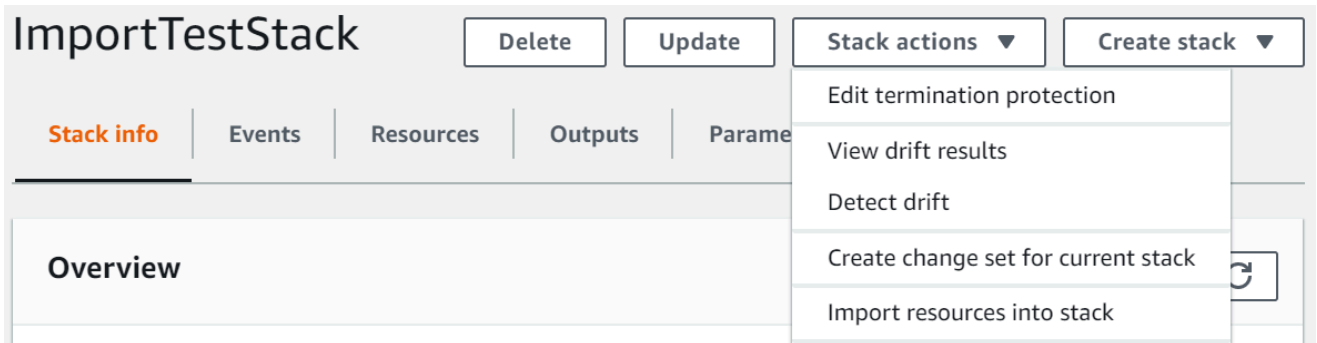
Example JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "Import test",
  "Resources": {
    "PlayersTable": {
      "Type": "AWS::DynamoDB::Table",
```

```
        "Properties": {
            "TableName": "Players",
            "AttributeDefinitions": [
                {
                    "AttributeName": "key",
                    "AttributeType": "S"
                }
            ],
            "KeySchema": [
                {
                    "AttributeName": "key",
                    "KeyType": "HASH"
                }
            ],
            "ProvisionedThroughput": {
                "ReadCapacityUnits": 5,
                "WriteCapacityUnits": 1
            }
        },
        "GamesTable": {
            "Type": "AWS::DynamoDB::Table",
            "DeletionPolicy": "Retain",
            "Properties": {
                "TableName": "Games",
                "AttributeDefinitions": [
                    {
                        "AttributeName": "key",
                        "AttributeType": "S"
                    }
                ],
                "KeySchema": [
                    {
                        "AttributeName": "key",
                        "KeyType": "HASH"
                    }
                ],
                "ProvisionedThroughput": {
                    "ReadCapacityUnits": 5,
                    "WriteCapacityUnits": 1
                }
            }
        }
    }
}
```

}

4. Ulangi langkah 2 — 3 untuk memperbarui tumpukan sumber lagi, kali ini untuk menghapus sumber daya target dari tumpukan.
5. Melakukan operasi impor untuk menambahkan GameTable ke tumpukan target.
 - a. Di halaman Tumpukan, dengan tumpukan induk yang dipilih, pilih Tindakan tumpukan, lalu pilih Impor sumber daya ke tumpukan.



- b. Baca halaman Gambaran umum Impor untuk daftar hal yang harus Anda sediakan selama operasi ini. Kemudian, pilih Selanjutnya.
- c. Pada halaman Tentukan template, lengkapi salah satu dari berikut ini, lalu pilih Berikutnya.
 - Pilih URL Amazon S3, lalu tentukan URL di kotak teks.
 - Pilih Unggah file templat, lalu telusuri file yang akan diunggah.
- d. Di halaman Identifikasi sumber daya, identifikasi sumber daya yang Anda pindahkan (dalam contoh ini, GameTable). Untuk informasi selengkapnya, lihat [Pengidentifikasi sumber daya](#).
 1. Di bawah Properti pengidentifikasi, pilih jenis pengidentifikasi sumber daya. Misalnya, sumber daya AWS::DynamoDB::Table dapat diidentifikasi menggunakan properti TableName.
 2. Di bawah Nilai pengidentifikasi, ketik nilai properti yang sebenarnya. Misalnya, *GameTables*.
 3. Pilih Berikutnya.
- e. Di halaman Tentukan detail tumpukan, ubah parameter apa pun, lalu pilih Selanjutnya. Ini secara otomatis membuat set perubahan.

⚠ Important

Operasi impor gagal jika Anda mengubah parameter yang sudah ada yang memulai operasi membuat, memperbarui, atau menghapus.

- f. Pada **TargetStackName** halaman Tinjauan, konfirmasi bahwa sumber daya yang benar sedang diimpor, lalu pilih Impor sumber daya. Ini secara otomatis memulai set perubahan yang dibuat pada langkah terakhir. Semua [tanda stack-level](#) diterapkan pada sumber daya yang diimpor saat ini.
- g. Panel Peristiwa di halaman Detail tumpukan menampilkan tumpukan induk Anda.

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

📌 Note

Tidak perlu menjalankan deteksi penyimpangan pada tumpukan induk setelah operasi impor ini karena sumber daya `AWS::CloudFormation::Stack` telah dikelola oleh CloudFormation.

Memfaktorkan ulang tumpukan menggunakan AWS CLI

1. Dalam template sumber, tentukan sumber daya `Retain` [DeletionPolicy](#) yang ingin Anda pindahkan.

Dalam contoh sumber templat berikut, `GameTable` adalah target faktor ulang (refactor) ini.

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. Perbarui tumpukan sumber untuk menerapkan kebijakan penghapusan ke sumber daya.

```
aws cloudformation update-stack --stack-name SourceStackName
```

3. Hapus sumber daya, parameter terkait, dan output dari templat sumber, lalu tambahkan ke templat target.

Templat sumber sekarang terlihat seperti berikut.

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
    }
}
}
}
}

```

Contoh templat target berikut saat ini memiliki sumber daya `PlayersTable`, dan sekarang juga berisi `GamesTable`.

Example JSON

```

{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "Import test",
  "Resources": {
    "PlayersTable": {
      "Type": "AWS::DynamoDB::Table",
      "Properties": {
        "TableName": "Players",
        "AttributeDefinitions": [
          {
            "AttributeName": "key",
            "AttributeType": "S"
          }
        ],
        "KeySchema": [
          {
            "AttributeName": "key",
            "KeyType": "HASH"
          }
        ],
        "ProvisionedThroughput": {
          "ReadCapacityUnits": 5,
          "WriteCapacityUnits": 1
        }
      }
    },
    "GamesTable": {
      "Type": "AWS::DynamoDB::Table",
      "DeletionPolicy": "Retain",
      "Properties": {

```



```

        "TableName": "Games",
        "AttributeDefinitions": [
            {
                "AttributeName": "key",
                "AttributeType": "S"
            }
        ],
        "KeySchema": [
            {
                "AttributeName": "key",
                "KeyType": "HASH"
            }
        ],
        "ProvisionedThroughput": {
            "ReadCapacityUnits": 5,
            "WriteCapacityUnits": 1
        }
    }
}
}
}

```

4. Perbarui tumpukan sumber untuk menghapus sumber daya GamesTable serta parameter dan output terkait dari tumpukan.

```
aws cloudformation update-stack --stack-name SourceStackName
```

5. Buat daftar sumber daya aktual untuk diimpor dan pengenal uniknya dalam format string JSON berikut. Untuk informasi selengkapnya, lihat [Pengidentifikasi sumber daya](#).

```
[{"ResourceType": "AWS::DynamoDB::Table", "LogicalResourceId": "GamesTable", "ResourceIdentifier": "GamesTable", "PhysicalResourceId": "GamesTable", "ResourceType": "AWS::DynamoDB::Table", "LogicalResourceId": "GamesTable", "ResourceIdentifier": "GamesTable", "PhysicalResourceId": "GamesTable"}]
```

Atau, Anda dapat menentukan parameter berformat JSON dalam file konfigurasi.

Misalnya, untuk mengimpor GamesTable, Anda dapat membuat *ResourcesToImport.txt* file yang berisi konfigurasi berikut.

```
[
{
    "ResourceType": "AWS::DynamoDB::Table",
    "LogicalResourceId": "GamesTable",

```

```

    "ResourceIdentifier": {
      "TableName": "Games"
    }
  }
]

```

6. Untuk membuat set perubahan, gunakan create-change-set perintah berikut dan ganti teks placeholder. Untuk --change-set-type opsi, tentukan nilai **IMPORT**. Untuk --resources-to-import opsi, ganti string JSON sampel dengan string JSON aktual yang baru saja Anda buat.

```

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 tidak mendukung YAMAL sebaris. Persyaratan untuk menghindari tanda kutip dalam string JSON bervariasi tergantung pada terminal Anda. Untuk informasi selengkapnya, lihat [Menggunakan tanda kutip di dalam string](#) di AWS Command Line Interface Panduan Pengguna.

Atau, Anda dapat menggunakan URL file sebagai masukan untuk --resources-to-import opsi, seperti yang ditunjukkan pada contoh berikut.

```

--resources-to-import file://ResourcesToImport.txt

```

7. Tinjau set perubahan untuk memastikan sumber daya yang benar sedang diimpor ke tumpukan target.

```

aws cloudformation describe-change-set \
  --change-set-name ImportChangeSet

```

8. Untuk memulai set perubahan dan mengimpor sumber daya, gunakan execute-change-set perintah berikut dan ganti teks placeholder. Semua tanda stack-level diterapkan ke sumber

daya yang diimpor pada saat ini. Setelah operasi (`IMPORT_COMPLETE`) berhasil diselesaikan, sumber daya berhasil diimpor.

```
aws cloudformation execute-change-set \  
  --change-set-name ImportChangeSet --stack-name TargetStackName
```

Note

Tidak perlu menjalankan deteksi penyimpangan pada tumpukan target setelah operasi impor ini karena sumber daya sudah dikelola oleh CloudFormation.

Membuat nest tumpukan yang ada

Gunakan fitur `resource import` untuk membuat nest tumpukan yang ada dalam tumpukan lain yang ada. Tumpukan bersarang adalah komponen umum yang Anda nyatakan dan merujuk dari dalam templat lain. Dengan begitu, Anda dapat menghindari menyalin dan menempelkan konfigurasi yang sama ke dalam templat dan menyederhanakan pembaruan tumpukan. Jika Anda memiliki templat untuk komponen umum, Anda dapat menggunakan sumber daya `AWS::CloudFormation::Stack` untuk merujuk templat ini dari dalam template lain. Untuk informasi selengkapnya tentang tumpukan bersarang, lihat [Pisahkan template menjadi potongan-potongan yang dapat digunakan kembali menggunakan tumpukan bersarang](#).

AWS CloudFormation hanya mendukung satu tingkat penggunaan `resource import` bersarang. Ini berarti Anda tidak dapat mengimpor tumpukan ke tumpukan anak atau mengimpor tumpukan yang memiliki anak.

Jika Anda baru mengimpor, kami sarankan Anda terlebih dahulu meninjau informasi pengantar dalam topik. [Impor AWS sumber daya ke CloudFormation tumpukan secara manual](#)

Validasi impor tumpukan bersarang

Selama operasi impor tumpukan bersarang, AWS CloudFormation lakukan validasi berikut.

- Definisi `AWS::CloudFormation::Stack` bersarang dalam templat tumpukan induk sesuai dengan templat tumpukan bersarang yang sebenarnya.
- Tanda untuk definisi `AWS::CloudFormation::Stack` bersarang dalam templat tumpukan induk sesuai dengan tanda untuk sumber daya tumpukan bersarang yang sebenarnya.

Membuat nest tumpukan yang ada menggunakan AWS Management Console

1. Tambahkan `AWS::CloudFormation::Stack` sumber daya ke template tumpukan induk dengan file Retain [DeletionPolicy](#). Dalam contoh berikut template tumpukan induk, `MyNestedStack` adalah target impor.

JSON

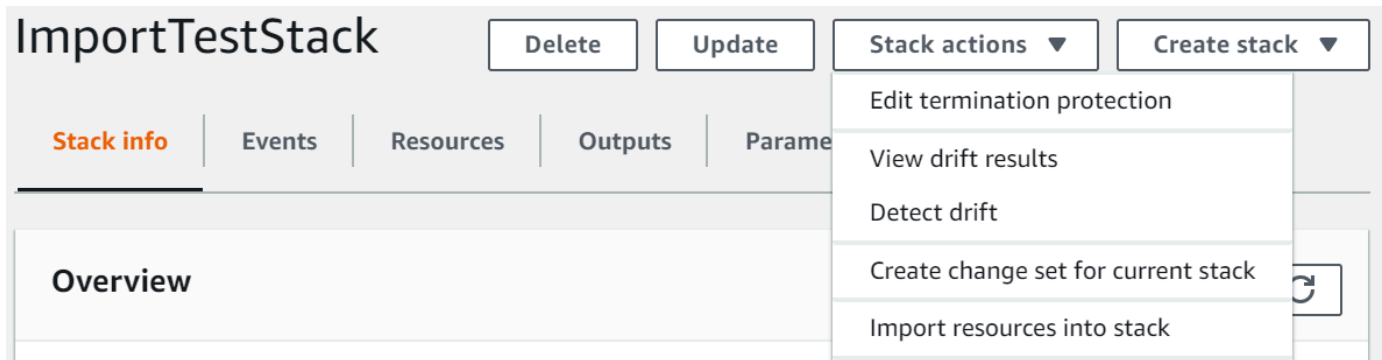
```
{
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Resources" : {
    "ServiceTable":{
      "Type":"AWS::DynamoDB::Table",
      "Properties":{
        "TableName":"Service",
        "AttributeDefinitions":[
          {
            "AttributeName":"key",
            "AttributeType":"S"
          }
        ],
        "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. Buka AWS CloudFormation konsol.
3. Di halaman Tumpukan, dengan tumpukan induk yang dipilih, pilih Tindakan tumpukan, lalu pilih Impor sumber daya ke tumpukan.



4. Baca halaman Gambaran umum Impor untuk daftar hal yang harus Anda sediakan selama operasi ini. Kemudian, pilih Selanjutnya.
5. Di halaman Tentukan templat, sediakan templat Anda yang telah diperbarui menggunakan salah satu metode berikut, lalu pilih Selanjutnya.
 - Pilih URL Amazon S3, lalu tentukan URL untuk templat Anda di kotak teks.
 - Pilih Unggah file templat, lalu telusuri templat Anda.
6. Di halaman Identifikasi sumber daya, identifikasi setiap sumber daya `AWS::CloudFormation::Stack`.
 - a. Di bawah Properti pengidentifikasi, pilih jenis pengidentifikasi sumber daya. Misalnya, sumber daya `AWS::CloudFormation::Stack` dapat diidentifikasi menggunakan properti `StackId`.
 - b. Di bawah nilai Identifier, ketik ARN dari tumpukan yang Anda impor. Misalnya, `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	Identifier value
StackId ▼	Enter StackId

- c. Pilih Berikutnya.

- Di halaman Tentukan detail tumpukan, ubah parameter apa pun, lalu pilih Selanjutnya. Ini secara otomatis membuat set perubahan.

Important

Operasi impor gagal jika Anda mengubah parameter yang sudah ada yang memulai operasi membuat, memperbarui, atau menghapus.

- Pada *MyParentStack* halaman Tinjauan, konfirmasi bahwa sumber daya yang benar sedang diimpor, lalu pilih Impor sumber daya. Ini secara otomatis menjalankan set perubahan yang dibuat pada langkah terakhir. Semua tanda stack-level diterapkan pada sumber daya yang diimpor saat ini.
- Panel Peristiwa di halaman Detail tumpukan menampilkan tumpukan induk Anda.

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	SamplelamUser	🟢 UPDATE_COMPLETE	-
2019-10-24 15:52:47 UTC-0700	SamplelamUser	🔄 UPDATE_IN_PROGRESS	Apply stack-level tags to imported resource if applicable.
2019-10-24 15:52:46 UTC-0700	SamplelamUser	🟢 IMPORT_COMPLETE	Resource import completed.
2019-10-24 15:52:45 UTC-0700	SamplelamUser	🔄 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

Tidak perlu menjalankan deteksi drift pada tumpukan induk setelah operasi impor ini karena `AWS::CloudFormation::Stack` sumber daya sudah dikelola oleh AWS CloudFormation.

Membuat nest tumpukan yang ada menggunakan AWS CLI

- Tambahkan `AWS::CloudFormation::Stack` sumber daya ke template tumpukan induk dengan file Retain [DeletionPolicy](#). Dalam contoh templat induk berikut, `MyNestedStack` adalah target impor.

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. Tulis string JSON seperti yang ditunjukkan pada contoh berikut, dengan modifikasi ini:

- Ganti *MyNestedStack* dengan ID logis dari sumber daya target seperti yang ditentukan dalam template.
- Ganti *arn:aws:cloudformation:us-west-2:12345678910:stack/mystack/5b918d10-cd98-11ea-90d5-0a9cd3354c10* dengan ARN dari tumpukan yang ingin Anda impor.

```

[{"ResourceType":"AWS::CloudFormation::Stack","LogicalResourceId":"MyNestedStack","Resource
{"StackId":"arn:aws:cloudformation:us-east-2:123456789012:stack/mystack/5b918d10-
cd98-11ea-90d5-0a9cd3354c10"}]}]

```

Atau, Anda dapat menentukan parameter dalam file konfigurasi.

Misalnya, untuk mengimpor `MyNestedStack`, Anda dapat membuat *ResourcesToImport.txt* file yang berisi konfigurasi berikut.

JSON

```
[
  {
    "ResourceType": "AWS::CloudFormation::Stack",
    "LogicalResourceId": "MyNestedStack",
    "ResourceIdentifier": {
      "StackId": "arn:aws:cloudformation:us-west-2:12345678910:stack/
mystack/5b918d10-cd98-11ea-90d5-0a9cd3354c10"
    }
  }
]
```

YAML

```
ResourceType: 'AWS::CloudFormation::Stack'
LogicalResourceId: MyNestedStack
ResourceIdentifier:
  StackId: >-
    arn:aws:cloudformation:us-west-2:12345678910:stack/mystack/5b918d10-
cd98-11ea-90d5-0a9cd3354c10
```

3. Untuk membuat set perubahan, gunakan `create-change-set` perintah berikut dan ganti teks placeholder. Untuk `--change-set-type` opsi, tentukan nilai **IMPORT**. Untuk `--resources-to-import` opsi, ganti string JSON sampel dengan string JSON aktual yang baru saja Anda buat.

```
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 tidak mendukung YAMAL sebaris. Persyaratan untuk menghindari tanda kutip dalam string JSON bervariasi tergantung pada terminal Anda. Untuk informasi selengkapnya, lihat [Menggunakan tanda kutip di dalam string](#) di AWS Command Line Interface Panduan Pengguna.

Atau, Anda dapat menggunakan URL file sebagai masukan untuk --resources-to-import opsi, seperti yang ditunjukkan pada contoh berikut.

```
--resources-to-import file:///ResourcesToImport.txt
```

Jika berhasil, perintah ini mengembalikan output sampel berikut.

```
{
  "Id": "arn:aws:cloudformation:us-west-2:12345678910:changeSet/
ImportChangeSet/8ad75b3f-665f-46f6-a200-0b4727a9442e",
  "StackId": "arn:aws:cloudformation:us-west-2:12345678910:stack/
MyParentStack/4e345b70-1281-11ef-b027-027366d8e82b"
}
```

4. Tinjau set perubahan untuk memastikan tumpukan yang benar sedang diimpor.

```
aws cloudformation describe-change-set --change-set-name ImportChangeSet
```

5. Untuk memulai set perubahan dan mengimpor tumpukan ke tumpukan induk sumber, gunakan execute-change-set perintah berikut dan ganti teks placeholder. Semua [tanda stack-level](#) diterapkan pada sumber daya yang diimpor saat ini. Setelah berhasil menyelesaikan operasi impor (IMPORT_COMPLETE), tumpukan berhasil dibuat nest-nya.

```
aws cloudformation execute-change-set --change-set-name ImportChangeSet
```

 Note

Tidak perlu menjalankan deteksi penyimpangan pada tumpukan induk setelah operasi impor ini karena sumber daya `AWS::CloudFormation::Stack` telah dikelola oleh AWS CloudFormation.

Impor AWS sumber daya ke CloudFormation tumpukan secara otomatis

Anda sekarang dapat mengimpor sumber daya bernama secara otomatis saat membuat atau memperbarui CloudFormation tumpukan. Sumber daya bernama adalah sumber dengan nama khusus. Untuk informasi selengkapnya, lihat [Jenis nama](#) di Referensi CloudFormation Template.

Saat Anda memulai impor otomatis, CloudFormation memeriksa sumber daya yang ada yang cocok dengan templat Anda dan mengimpornya selama penerapan. Untuk tumpukan bersarang, buat set perubahan dari tumpukan root.

Setelah impor selesai dan sebelum melakukan operasi tumpukan berikutnya, kami sarankan untuk menjalankan deteksi drift pada sumber daya impor. Deteksi drift memastikan bahwa konfigurasi templat cocok dengan konfigurasi yang sebenarnya. Untuk informasi selengkapnya, lihat [Mendeteksi penyimpangan di seluruh tumpukan CloudFormation](#).

Untuk mengimpor sumber daya, mereka harus memenuhi persyaratan berikut:

- Sumber daya harus memiliki nama kustom statis yang ditentukan dalam template Anda. Nama dinamis (menggunakan `!Ref` atau fungsi lainnya) saat ini tidak didukung.
- Sumber daya harus memiliki `DeletionPolicy` dari `Retain` atau `RetainExceptOnCreate`.
- Sumber daya harus belum menjadi milik CloudFormation tumpukan lain.
- Jenis sumber daya harus mendukung operasi CloudFormation impor. Untuk informasi selengkapnya, lihat [Dukungan jenis sumber daya](#).
- ID utama untuk jenis sumber daya harus ada di template. Primer IDs dengan properti hanya baca tidak didukung. Untuk mengetahui apa ID utama untuk suatu tipe, cari `primaryIdentifier` properti dalam skema sumber daya. Untuk informasi selengkapnya tentang properti, lihat [PrimaryIdentifier](#).

Example Contoh impor otomatis

Contoh berikut menggunakan set perubahan, `CreateChangeSet` untuk membuat tumpukan yang disebut `my-stack` berdasarkan file templat `template.yaml`, dan mengimpor sumber daya yang cocok secara otomatis.

```
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
```

Pemecahan Masalah

Jika impor otomatis gagal, lakukan hal berikut untuk memecahkan masalah:

- Verifikasi nama sumber daya di template Anda sama persis dengan nama sumber daya
- Verifikasi bahwa sumber daya belum dikelola oleh tumpukan lain
- Pastikan jenis sumber daya mendukung operasi impor
- Verifikasi template Anda mencakup semua properti yang diperlukan untuk jenis sumber daya

Pengembalian operasi impor

Untuk mengembalikan operasi impor, tentukan kebijakan penghapusan `Retain` untuk sumber daya yang ingin dihapus dari templat untuk memastikan agar ia dipertahankan saat Anda menghapusnya dari tumpukan.

Mengembalikan operasi impor menggunakan AWS Management Console

1. Tentukan sumber daya yang ingin Anda hapus dari tumpukan Anda. Retain [DeletionPolicy](#)
Pada templat contoh berikut, `GameTable` adalah target dari operasi pengembalian ini.

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. Buka CloudFormation konsol untuk melakukan pembaruan tumpukan guna menerapkan kebijakan penghapusan.
 - a. Pada halaman Tumpukan, dengan tumpukan dipilih, pilih Perbarui, lalu pilih Perbarui tumpukan (standar).
 - b. Di bawah Siapkan templat, pilih Ganti templat saat ini.
 - c. Di bawah Tentukan templat, sediakan templat sumber yang diperbarui dengan atribut `DeletionPolicy` pada `GameTable`, lalu pilih Selanjutnya.
 - Pilih URL Amazon S3, lalu tentukan URL ke templat sumber yang diperbarui di kotak teks.
 - Pilih Unggah file templat, lalu telusuri file templat sumber yang diperbarui.
 - d. Pada halaman Tentukan detail tumpukan, tidak ada perubahan yang diperlukan. Pilih Berikutnya.
 - e. Pada halaman Konfigurasi pilihan tumpukan, tidak ada perubahan yang diperlukan. Pilih Berikutnya.
 - f. Pada *MyStack* halaman Ulasan, tinjau perubahan Anda. Jika template Anda berisi sumber daya IAM, pilih Saya mengakui bahwa template ini dapat membuat sumber daya IAM untuk menentukan bahwa Anda ingin menggunakan sumber daya IAM dalam template. Untuk informasi selengkapnya, lihat [Mengakui sumber daya IAM dalam templat CloudFormation](#). Kemudian, perbarui tumpukan sumber Anda dengan membuat set perubahan atau perbarui tumpukan sumber Anda secara langsung.
3. Hapus sumber daya, parameter terkait, dan output dari templat tumpukan. Dalam contoh ini, templat sekarang terlihat seperti berikut.

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. Ulangi langkah 2 untuk menghapus sumber daya (GameTable) serta parameter terkait dan output dari tumpukan.

Mengembalikan operasi impor menggunakan AWS CLI

1. Tentukan sumber daya yang ingin Anda hapus dari tumpukan Anda. Retain [DeletionPolicy](#) Pada templat contoh berikut, GameTable adalah target dari operasi pengembalian ini.

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. Perbarui tumpukan untuk menerapkan kebijakan penghapusan ke sumber daya.

```
aws cloudformation update-stack --stack-name MyStack
```

3. Hapus sumber daya, parameter terkait, dan output dari templat tumpukan. Dalam contoh ini, templat sekarang terlihat seperti berikut.

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. Perbarui tumpukan untuk menghapus sumber daya (GameTable) serta parameter terkait dan output dari tumpukan.

```
aws cloudformation update-stack --stack-name MyStack
```

Stack refactoring

Stack refactoring menyederhanakan penataan ulang sumber daya di CloudFormation tumpukan Anda sambil tetap mempertahankan properti dan data sumber daya yang ada. Dengan stack refactoring, Anda dapat memindahkan sumber daya antar tumpukan, membagi tumpukan monolitik menjadi komponen yang lebih kecil, atau mengkonsolidasikan beberapa tumpukan menjadi satu.

Cara kerja stack refactoring

Pertimbangkan langkah-langkah di bawah ini saat merencanakan refactor tumpukan Anda:

1. Nilai infrastruktur Anda saat ini: Tinjau CloudFormation tumpukan dan sumber daya yang ada untuk mengidentifikasi peluang pemfaktoran ulang tumpukan.
2. Rencanakan refactor Anda: Tentukan bagaimana sumber daya harus diatur. Pertimbangkan dependensi, konvensi penamaan, dan batasan operasional Anda. Ini dapat mempengaruhi CloudFormation validasi nanti.

Tentukan jumlah tumpukan tujuan tempat Anda akan memfaktorkan ulang sumber daya. Anda dapat memindahkan sumber daya antara setidaknya 2 tumpukan, dan maksimal 5 tumpukan. Sumber daya dapat dipindahkan di antara tumpukan bersarang.

3. Perbarui templat Anda: Ubah CloudFormation templat Anda untuk mencerminkan perubahan yang direncanakan, seperti memindahkan definisi sumber daya antar templat. Anda dapat mengganti nama logis IDs selama refactoring.
4. Buat refactor tumpukan: Berikan daftar nama tumpukan dan templat yang ingin Anda refactor.
5. Tinjau dampak refactor pada infrastruktur Anda dan selesaikan konflik apa pun: CloudFormation memvalidasi template yang Anda berikan dan memeriksa dependensi lintas tumpukan, jenis sumber daya dengan masalah pembaruan tag, dan konflik ID logis sumber daya.

Jika validasi berhasil, CloudFormation akan menghasilkan pratinjau tindakan refactor yang akan terjadi setelah mengeksekusi refactor.

Jika validasi gagal, Anda dapat mencoba lagi setelah menyelesaikan masalah yang ditemukan. Jika ada konflik, Anda perlu menyediakan pemetaan ID logis sumber daya yang menunjukkan sumber dan tujuan sumber daya dalam konflik.

6. Jalankan refactor: Setelah mengonfirmasi perubahan sejajar dengan cara Anda ingin memfaktorkan ulang tumpukan Anda, jalankan refactor tumpukan.
7. Monitor: Lihat status stack refactor untuk memastikannya berhasil. `ExecutionStatus`

Batasan refactoring tumpukan

Pertimbangkan batasan berikut saat merencanakan refactor tumpukan Anda:

- Operasi refactor tidak mengizinkan pembuatan sumber daya baru, penghapusan sumber daya, atau perubahan konfigurasi sumber daya.
- Anda tidak dapat mengubah atau menambahkan parameter, kondisi, atau pemetaan baru selama refactor tumpukan. Solusi potensial adalah memperbarui tumpukan Anda sebelum melakukan refactor.
- Anda tidak dapat memfaktorkan ulang sumber daya yang sama menjadi beberapa tumpukan.
- Anda tidak dapat memfaktorkan ulang sumber daya ke tumpukan baru jika tumpukan itu menggunakan parameter semu tertentu (Contoh: `AWS::StackName`).
- CloudFormation tidak mendukung tumpukan kosong. Sebelum membuat refactor tumpukan yang akan menghapus semua sumber daya yang ada dari tumpukan, Anda harus menambahkan sumber daya ke tumpukan itu. Sumber daya ini bisa sederhana, seperti tipe `waitCondition` sumber daya.
- Stack refactor tidak mendukung tumpukan yang berisi kebijakan tumpukan, termasuk kebijakan yang mengizinkan perubahan sumber daya.
- Sumber daya berikut tidak tersedia:
 - `AWS::ACMPCA::Certificate`
 - `AWS::ACMPCA::CertificateAuthority`
 - `AWS::ACMPCA::CertificateAuthorityActivation`
 - `AWS::ApiGateway::BasePathMapping`
 - `AWS::ApiGateway::Method`
 - `AWS::AppConfig::ConfigurationProfile`
 - `AWS::AppConfig::Deployment`
 - `AWS::AppConfig::Environment`
 - `AWS::AppConfig::Extension`
 - `AWS::AppConfig::ExtensionAssociation`
 - `AWS::AppStream::DirectoryConfig`
 - `AWS::AppStream::StackFleetAssociation`
 - `AWS::AppStream::StackUserAssociation`
 - `AWS::AppStream::User`

- `AWS::BackupGateway::Hypervisor`
- `AWS::CodePipeline::CustomActionType`
- `AWS::Cognito::UserPoolRiskConfigurationAttachment`
- `AWS::Cognito::UserPoolUICustomizationAttachment`
- `AWS::Cognito::UserPoolUserToGroupAttachment`
- `AWS::Config::ConfigRule`
- `AWS::DataBrew::Dataset`
- `AWS::DataBrew::Job`
- `AWS::DataBrew::Project`
- `AWS::DataBrew::Recipe`
- `AWS::DataBrew::Ruleset`
- `AWS::DataBrew::Schedule`
- `AWS::DataZone::DataSource`
- `AWS::DataZone::Environment`
- `AWS::DataZone::EnvironmentBlueprintConfiguration`
- `AWS::DataZone::EnvironmentProfile`
- `AWS::DataZone::Project`
- `AWS::DataZone::SubscriptionTarget`
- `AWS::DynamoDB::GlobalTable`
- `AWS::EC2::LaunchTemplate`
- `AWS::EC2::SpotFleet`
- `AWS::EC2::VolumeAttachment`
- `AWS::EC2::VPCDHCPOptionsAssociation`
- `AWS::ElasticBeanstalk::ConfigurationTemplate`
- `AWS::FIS::ExperimentTemplate`
- `AWS::Glue::Schema`
- `AWS::GuardDuty::IPSet`
- `AWS::GuardDuty::PublishingDestination`
- `AWS::GuardDuty::ThreatIntelSet`
- `AWS::ImageBuilder::Component`

- `AWS::IoT::FleetWise::Campaign`
- `AWS::IoT::Wireless::WirelessDeviceImportTask`
- `AWS::Lambda::EventInvokeConfig`
- `AWS::Lex::BotVersion`
- `AWS::M2::Application`
- `AWS::Maester::DocumentType`
- `AWS::MediaTailor::Channel`
- `AWS::MSK::Configuration`
- `AWS::MSK::ServerlessCluster`
- `AWS::NeptuneGraph::PrivateGraphEndpoint`
- `AWS::Omics::AnnotationStore`
- `AWS::Omics::ReferenceStore`
- `AWS::Omics::SequenceStore`
- `AWS::OpenSearchServerless::Collection`
- `AWS::Panorama::PackageVersion`
- `AWS::PCAConnectorAD::Connector`
- `AWS::PCAConnectorAD::DirectoryRegistration`
- `AWS::PCAConnectorAD::Template`
- `AWS::PCAConnectorAD::TemplateGroupAccessControlEntry`
- `AWS::QuickSight::Theme`
- `AWS::RefactorSpaces::Environment`
- `AWS::RefactorSpaces::Route`
- `AWS::RefactorSpaces::Service`
- `AWS::RoboMaker::RobotApplication`
- `AWS::RoboMaker::SimulationApplication`
- `AWS::SageMaker::InferenceComponent`
- `AWS::ServiceCatalog::PortfolioPrincipalAssociation`
- `AWS::ServiceCatalog::PortfolioProductAssociation`
- `AWS::ServiceCatalog::PortfolioShare`
- `AWS::ServiceCatalog::TagOptionAssociation`

- `AWS::ServiceCatalogAppRegistry::AttributeGroupAssociation`
- `AWS::ServiceCatalogAppRegistry::ResourceAssociation`
- `AWS::StepFunctions::StateMachineVersion`
- `AWS::Synthetics::Canary`
- `AWS::VoiceID::Domain`
- `AWS::WAFv2::IPSet`
- `AWS::WAFv2::RegexPatternSet`
- `AWS::WAFv2::RuleGroup`
- `AWS::WAFv2::WebACL`

Memfaktorkan ulang tumpukan menggunakan AWS Command Line Interface

Perintah berikut digunakan untuk stack refactoring:

- [create-stack-refactor](#)
- [describe-stack-refactor](#)
- [execute-stack-refactor](#)
- [list-stack-refactors](#)
- [list-stack-refactor-actions](#)

Di bawah ini adalah contoh cara memfaktorkan ulang tumpukan menggunakan AWS Command Line Interface (CLI).

1. Untuk memulai, Anda akan memerlukan CloudFormation template yang ingin Anda refactor. Perintah berikut mengambil template:

```
aws cloudformation get-template --stack-name ExampleStack1
```

Setelah Anda memiliki template, gunakan lingkungan pengembangan terintegrasi (IDE) pilihan Anda untuk memperbaruinya untuk menggunakan struktur dan organisasi sumber daya yang diinginkan.

2. Buat stack refactor menggunakan `create-stack-refactor` perintah, dan berikan nama tumpukan dan template untuk setiap tumpukan yang terlibat dalam refactoring:

```
aws cloudformation create-stack-refactor \  
  --stack-definitions \  
    StackName=MySns,TemplateBody@=file://afterSns.yaml \  
    StackName=MyLambdaSubscription,TemplateBody@=file://afterLambda.yaml \  
  --enable-stack-creation \  
  --resource-mappings file://refactor.json
```

`--resource-mappingsParameter` ini opsional, tetapi akan diperlukan jika konflik terdeteksi selama validasi template. Selain menyediakan Source dan Destination tumpukan, Anda juga perlu menyediakan. `LogicalResourceId` Berikut ini adalah contoh `refactor.json` file.

```
[  
  {  
    "Source": {  
      "StackName": "MySns",  
      "LogicalResourceId": "MyFunction"  
    },  
    "Destination": {  
      "StackName": "MyLambdaSubscription",  
      "LogicalResourceId": "Function"  
    }  
  }  
]
```

3. Setelah membuat refactor, CloudFormation buat pratinjau perubahan refactor, yang memungkinkan Anda melihat pratinjau dampak perubahan yang diusulkan terhadap infrastruktur Anda. Dalam contoh ini, yang `StackRefactorId` dibuat adalah `stack-refactor-1ab2-c34d-5ef6`. Gunakan perintah berikut untuk melihat pratinjau perubahan:

```
aws cloudformation list-stack-refactor-actions \  
  --stack-refactor-id stack-refactor-1ab2-c34d-5ef6
```

4. Setelah meninjau dan mengonfirmasi perubahan Anda, gunakan `execute-stack-refactor` perintah untuk menyelesaikan proses refactoring:

```
aws cloudformation execute-stack-refactor \  
  --stack-refactor-id stack-refactor-1ab2-c34d-5ef6
```


5. Setelah mengeksekusi `execute-stack-refactor`, Anda dapat memantau status operasi menggunakan perintah berikut:

```
aws cloudformation describe-stack-refactor \  
--stack-refactor-id stack-refactor-1ab2-c34d-5ef6
```

CloudFormation akan secara otomatis memperbarui Status dan ExecutionStatus operasi refactor.

Dukungan jenis sumber daya

Tabel berikut mencantumkan jenis AWS sumber daya yang saat ini mendukung operasi generator impor, deteksi drift, dan infrastruktur sebagai kode (IAC). Setiap nama jenis sumber daya menautkan ke topik referensi yang sesuai di [Panduan Referensi AWS CloudFormation Template](#).

Sumber daya yang mendukung impor sumber daya juga dapat mendukung impor otomatis. Untuk informasi selengkapnya, lihat [Impor AWS sumber daya ke CloudFormation tumpukan](#).

Ini bukan daftar sumber daya yang lengkap. AWS Jika jenis sumber daya tertentu tidak terdaftar, itu mungkin berarti bahwa itu tidak dapat diakses melalui AWS Cloud Control API. Untuk informasi selengkapnya, lihat [Jenis sumber daya yang mendukung Cloud Control API](#) di Panduan Pengguna Cloud Control API. Setiap AWS layanan secara independen menentukan jenis sumber daya mana yang dapat diakses melalui Cloud Control API.

CloudFormation juga mendukung operasi deteksi impor dan drift untuk jenis sumber daya pribadi yang dapat disediakan (yang memiliki jenis penyediaan salah satu atau). `FULLY_MUTABLE` `IMMUTABLE` Untuk mengimpor atau melakukan deteksi drift pada jenis sumber daya pribadi, Anda harus terlebih dahulu mendaftarkan versi default dari jenis sumber daya tersebut di akun Anda, dan memastikannya dapat disediakan. Untuk informasi selengkapnya, lihat [Gunakan ekstensi pribadi pihak ketiga yang telah dibagikan dengan Anda](#).

Perhatikan bahwa generator IAC hanya mendukung AWS sumber daya yang kompatibel dengan Cloud Control API di Wilayah Anda.

Untuk mengakses informasi secara terprogram tentang jenis sumber daya publik dan pribadi yang dapat disediakan, Anda dapat menggunakan AWS Cloud Control API Untuk informasi selengkapnya, lihat [Menentukan apakah jenis sumber daya mendukung Cloud Control API](#) di Panduan Pengguna Cloud Control API.

Untuk memulai impor, deteksi drift, atau generator IAc, berikut adalah beberapa topik yang berguna untuk ditinjau:

- [Impor AWS sumber daya ke CloudFormation tumpukan](#)
- [Mendeteksi perubahan konfigurasi yang tidak terkelola pada tumpukan dan sumber daya dengan deteksi drift](#)
- [Hasilkan templat dari sumber daya yang ada dengan generator IAc](#)

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
AWS::ACMPCA::Certificate	 Ya	 Tidak	 Tidak
AWS::ACMPCA::CertificateAuthority	 Ya	 Ya	 Ya
AWS::ACMPCA::CertificateAuthorityActivation	 Ya	 Tidak	 Tidak
AWS::ACMPCA::Permission	 Ya	 Ya	 Tidak
AWS::APS::RuleGroupsNamespace	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::APS::Scraper</u>	 Ya	 Ya	 Tidak
<u>AWS::APS::Workspace</u>	 Ya	 Ya	 Ya
<u>AWS::ARCZonalShift::AutoshiftObserve rNotificationStatus</u>	 Ya	 Ya	 Tidak
<u>AWS::ARCZonalShift::ZonalAutoshiftCo nfiguration</u>	 Ya	 Ya	 Tidak
<u>AWS::AccessAnalyzer::Analyzer</u>	 Ya	 Ya	 Ya
<u>AWS::AmazonMQ::Configuration</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::Amplify::App</u>	 Ya	 Ya	 Ya
<u>AWS::Amplify::Branch</u>	 Ya	 Ya	 Ya
<u>AWS::Amplify::Domain</u>	 Ya	 Ya	 Ya
<u>AWS::AmplifyUIBuilder::Component</u>	 Ya	 Ya	 Ya
<u>AWS::AmplifyUIBuilder::Form</u>	 Ya	 Ya	 Ya
<u>AWS::AmplifyUIBuilder::Theme</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::ApiGateway::Account</u>	 Ya	 Ya	 Tidak
<u>AWS::ApiGateway::ApiKey</u>	 Ya	 Ya	 Ya
<u>AWS::ApiGateway::Authorizer</u>	 Ya	 Ya	 Tidak
<u>AWS::ApiGateway::BasePathMapping</u>	 Ya	 Ya	 Tidak
<u>AWS::ApiGateway::BasePathMappingV2</u>	 Ya	 Ya	 Tidak
<u>AWS::ApiGateway::ClientCertificate</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::ApiGateway::Deployment</u>	 Ya	 Ya	 Ya
<u>AWS::ApiGateway::DocumentationPart</u>	 Ya	 Ya	 Tidak
<u>AWS::ApiGateway::DocumentationVersion</u>	 Ya	 Ya	 Ya
<u>AWS::ApiGateway::DomainName</u>	 Ya	 Ya	 Tidak
<u>AWS::ApiGateway::DomainNameAccessAssociation</u>	 Ya	 Ya	 Tidak
<u>AWS::ApiGateway::DomainNameV2</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::ApiGateway::GatewayResponse</u>	 Ya	 Ya	 Tidak
<u>AWS::ApiGateway::Method</u>	 Ya	 Ya	 Tidak
<u>AWS::ApiGateway::Model</u>	 Ya	 Ya	 Tidak
<u>AWS::ApiGateway::RequestValidator</u>	 Ya	 Ya	 Tidak
<u>AWS::ApiGateway::Resource</u>	 Ya	 Ya	 Tidak
<u>AWS::ApiGateway::RestApi</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::ApiGateway::Stage</u>	 Ya	 Tidak	 Ya
<u>AWS::ApiGateway::UsagePlan</u>	 Ya	 Ya	 Ya
<u>AWS::ApiGateway::UsagePlanKey</u>	 Ya	 Tidak	 Ya
<u>AWS::ApiGateway::VpcLink</u>	 Ya	 Ya	 Tidak
<u>AWS::ApiGatewayV2::Api</u>	 Ya	 Ya	 Ya
<u>AWS::ApiGatewayV2::ApiMapping</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::ApiGatewayV2::Authorizer</u>	 Ya	 Ya	 Tidak
<u>AWS::ApiGatewayV2::Deployment</u>	 Ya	 Ya	 Ya
<u>AWS::ApiGatewayV2::DomainName</u>	 Ya	 Ya	 Ya
<u>AWS::ApiGatewayV2::Integration</u>	 Ya	 Ya	 Tidak
<u>AWS::ApiGatewayV2::IntegrationResponse</u>	 Ya	 Ya	 Tidak
<u>AWS::ApiGatewayV2::Model</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::ApiGatewayV2::Route</u>	 Ya	 Ya	 Tidak
<u>AWS::ApiGatewayV2::RouteResponse</u>	 Ya	 Ya	 Tidak
<u>AWS::ApiGatewayV2::RoutingRule</u>	 Ya	 Ya	 Tidak
<u>AWS::ApiGatewayV2::VpcLink</u>	 Ya	 Ya	 Ya
<u>AWS::AppConfig::Application</u>	 Ya	 Ya	 Ya
<u>AWS::AppConfig::ConfigurationProfile</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::AppConfig::Deployment</u>	 Ya	 Ya	 Tidak
<u>AWS::AppConfig::DeploymentStrategy</u>	 Ya	 Ya	 Tidak
<u>AWS::AppConfig::Environment</u>	 Ya	 Ya	 Tidak
<u>AWS::AppConfig::Extension</u>	 Ya	 Ya	 Tidak
<u>AWS::AppConfig::ExtensionAssociation</u>	 Ya	 Ya	 Tidak
<u>AWS::AppConfig::HostedConfigurationVersion</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAC
<u>AWS::AppFlow::Connector</u>	 Ya	 Ya	 Ya
<u>AWS::AppFlow::ConnectorProfile</u>	 Ya	 Ya	 Ya
<u>AWS::AppFlow::Flow</u>	 Ya	 Ya	 Ya
<u>AWS::AppIntegrations::Application</u>	 Ya	 Ya	 Tidak
<u>AWS::AppIntegrations::DataIntegration</u>	 Ya	 Ya	 Ya
<u>AWS::AppIntegrations::EventIntegration</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::AppRunner::AutoScalingConfiguration</u>	 Ya	 Ya	 Tidak
<u>AWS::AppRunner::ObservabilityConfiguration</u>	 Ya	 Ya	 Tidak
<u>AWS::AppRunner::Service</u>	 Ya	 Ya	 Tidak
<u>AWS::AppRunner::VpcConnector</u>	 Ya	 Ya	 Tidak
<u>AWS::AppRunner::VpcIngressConnection</u>	 Ya	 Ya	 Tidak
<u>AWS::AppStream::AppBlock</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::AppStream::AppBlockBuilder</u>	 Ya	 Ya	 Tidak
<u>AWS::AppStream::Application</u>	 Ya	 Ya	 Tidak
<u>AWS::AppStream::ApplicationEntitlementAssociation</u>	 Ya	 Ya	 Tidak
<u>AWS::AppStream::ApplicationFleetAssociation</u>	 Ya	 Ya	 Tidak
<u>AWS::AppStream::DirectoryConfig</u>	 Ya	 Ya	 Ya
<u>AWS::AppStream::Entitlement</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::AppStream::ImageBuilder</u>	 Ya	 Ya	 Ya
<u>AWS::AppSync::Api</u>	 Ya	 Ya	 Tidak
<u>AWS::AppSync::ChannelNamespace</u>	 Ya	 Ya	 Tidak
<u>AWS::AppSync::DataSource</u>	 Ya	 Ya	 Tidak
<u>AWS::AppSync::DomainName</u>	 Ya	 Ya	 Ya
<u>AWS::AppSync::DomainNameApiAssociation</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::AppSync::FunctionConfiguration</u>	 Ya	 Ya	 Tidak
<u>AWS::AppSync::GraphQLApi</u>	 Ya	 Ya	 Tidak
<u>AWS::AppSync::Resolver</u>	 Ya	 Ya	 Tidak
<u>AWS::AppSync::SourceApiAssociation</u>	 Ya	 Ya	 Ya
<u>AWS::ApplicationAutoScaling::ScalableTarget</u>	 Ya	 Ya	 Ya
<u>AWS::ApplicationAutoScaling::ScalingPolicy</u>	 Ya	 Tidak	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::ApplicationInsights::Application</u>	 Ya	 Ya	 Tidak
<u>AWS::ApplicationSignals::Discovery</u>	 Ya	 Ya	 Tidak
<u>AWS::ApplicationSignals::ServiceLevelObjective</u>	 Ya	 Ya	 Tidak
<u>AWS::Athena::CapacityReservation</u>	 Ya	 Ya	 Ya
<u>AWS::Athena::DataCatalog</u>	 Ya	 Ya	 Ya
<u>AWS::Athena::NamedQuery</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::Athena::PreparedStatement</u>	 Ya	 Ya	 Ya
<u>AWS::Athena::WorkGroup</u>	 Ya	 Ya	 Ya
<u>AWS::AuditManager::Assessment</u>	 Ya	 Tidak	 Tidak
<u>AWS::AutoScaling::AutoScalingGroup</u>	 Ya	 Tidak	 Ya
<u>AWS::AutoScaling::LaunchConfiguration</u>	 Ya	 Ya	 Ya
<u>AWS::AutoScaling::LifecycleHook</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::AutoScaling::ScalingPolicy</u>	 Ya	 Ya	 Ya
<u>AWS::AutoScaling::ScheduledAction</u>	 Ya	 Ya	 Ya
<u>AWS::AutoScaling::WarmPool</u>	 Ya	 Ya	 Tidak
<u>AWS::B2BI::Capability</u>	 Ya	 Ya	 Tidak
<u>AWS::B2BI::Partnership</u>	 Ya	 Ya	 Tidak
<u>AWS::B2BI::Profile</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::B2BI::Transformer</u>	 Ya	 Ya	 Tidak
<u>AWS::Backup::BackupPlan</u>	 Ya	 Ya	 Tidak
<u>AWS::Backup::BackupSelection</u>	 Ya	 Tidak	 Ya
<u>AWS::Backup::BackupVault</u>	 Ya	 Ya	 Tidak
<u>AWS::Backup::Framework</u>	 Ya	 Ya	 Ya
<u>AWS::Backup::LogicallyAirGappedBackupVault</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::Backup::ReportPlan</u>	 Ya	 Ya	 Ya
<u>AWS::Backup::RestoreTestingPlan</u>	 Ya	 Ya	 Tidak
<u>AWS::Backup::RestoreTestingSelection</u>	 Ya	 Ya	 Tidak
<u>AWS::BackupGateway::Hypervisor</u>	 Ya	 Ya	 Tidak
<u>AWS::Batch::ComputeEnvironment</u>	 Ya	 Ya	 Ya
<u>AWS::Batch::ConsumableResource</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::Batch::JobDefinition</u>	 Ya	 Ya	 Tidak
<u>AWS::Batch::JobQueue</u>	 Ya	 Ya	 Ya
<u>AWS::Batch::SchedulingPolicy</u>	 Ya	 Ya	 Ya
<u>AWS::Bedrock::Agent</u>	 Ya	 Ya	 Tidak
<u>AWS::Bedrock::AgentAlias</u>	 Ya	 Ya	 Tidak
<u>AWS::Bedrock::ApplicationInferenceProfile</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::Bedrock::Blueprint</u>	 Ya	 Ya	 Tidak
<u>AWS::Bedrock::DataAutomationProject</u>	 Ya	 Ya	 Tidak
<u>AWS::Bedrock::DataSource</u>	 Ya	 Ya	 Tidak
<u>AWS::Bedrock::Flow</u>	 Ya	 Ya	 Tidak
<u>AWS::Bedrock::FlowAlias</u>	 Ya	 Ya	 Tidak
<u>AWS::Bedrock::FlowVersion</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::Bedrock::Guardrail</u>	 Ya	 Ya	 Tidak
<u>AWS::Bedrock::GuardrailVersion</u>	 Ya	 Ya	 Tidak
<u>AWS::Bedrock::IntelligentPromptRouter</u>	 Ya	 Ya	 Tidak
<u>AWS::Bedrock::KnowledgeBase</u>	 Ya	 Tidak	 Tidak
<u>AWS::Bedrock::Prompt</u>	 Ya	 Ya	 Tidak
<u>AWS::Bedrock::PromptVersion</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::Budgets::BudgetsAction</u>	 Ya	 Ya	 Ya
<u>AWS::CE::AnomalyMonitor</u>	 Ya	 Ya	 Tidak
<u>AWS::CE::AnomalySubscription</u>	 Ya	 Ya	 Tidak
<u>AWS::CE::CostCategory</u>	 Ya	 Ya	 Ya
<u>AWS::Cassandra::Keyspace</u>	 Ya	 Ya	 Tidak
<u>AWS::Cassandra::Table</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::Cassandra::Type</u>	 Ya	 Ya	 Tidak
<u>AWS::CertificateManager::Account</u>	 Ya	 Ya	 Tidak
<u>AWS::Chatbot::CustomAction</u>	 Ya	 Ya	 Tidak
<u>AWS::Chatbot::MicrosoftTeamsChannelConfiguration</u>	 Ya	 Ya	 Ya
<u>AWS::Chatbot::SlackChannelConfiguration</u>	 Ya	 Ya	 Ya
<u>AWS::CleanRooms::AnalysisTemplate</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::CleanRooms::Collaboration</u>	 Ya	 Ya	 Ya
<u>AWS::CleanRooms::ConfiguredTable</u>	 Ya	 Ya	 Ya
<u>AWS::CleanRooms::ConfiguredTableAssociation</u>	 Ya	 Ya	 Ya
<u>AWS::CleanRooms::IdMappingTable</u>	 Ya	 Ya	 Tidak
<u>AWS::CleanRooms::IdNamespaceAssociation</u>	 Ya	 Ya	 Tidak
<u>AWS::CleanRooms::Membership</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::CleanRooms::PrivacyBudgetTemplate</u>	 Ya	 Ya	 Tidak
<u>AWS::CleanRoomsML::TrainingDataset</u>	 Ya	 Ya	 Tidak
<u>AWS::CloudFormation::GuardHook</u>	 Ya	 Ya	 Tidak
<u>AWS::CloudFormation::HookDefaultVersion</u>	 Ya	 Ya	 Ya
<u>AWS::CloudFormation::HookTypeConfig</u>	 Ya	 Ya	 Ya
<u>AWS::CloudFormation::HookVersion</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::CloudFormation::LambdaHook</u>	 Ya	 Ya	 Tidak
<u>AWS::CloudFormation::ModuleDefaultVersion</u>	 Ya	 Ya	 Tidak
<u>AWS::CloudFormation::ModuleVersion</u>	 Ya	 Ya	 Tidak
<u>AWS::CloudFormation::PublicTypeVersion</u>	 Ya	 Ya	 Tidak
<u>AWS::CloudFormation::Publisher</u>	 Ya	 Ya	 Tidak
<u>AWS::CloudFormation::ResourceDefaultVersion</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::CloudFormation::ResourceVersion</u>	 Ya	 Ya	 Tidak
<u>AWS::CloudFormation::Stack</u>	 Ya	 Ya	 Tidak
<u>AWS::CloudFormation::StackSet</u>	 Ya	 Ya	 Tidak
<u>AWS::CloudFormation::TypeActivation</u>	 Ya	 Ya	 Tidak
<u>AWS::CloudFront::AnycastIpList</u>	 Ya	 Ya	 Tidak
<u>AWS::CloudFront::CachePolicy</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::CloudFront::CloudFrontOriginAccessIdentity</u>	 Ya	 Ya	 Ya
<u>AWS::CloudFront::ConnectionGroup</u>	 Ya	 Ya	 Tidak
<u>AWS::CloudFront::ContinuousDeploymentPolicy</u>	 Ya	 Ya	 Ya
<u>AWS::CloudFront::Distribution</u>	 Ya	 Ya	 Ya
<u>AWS::CloudFront::DistributionTenant</u>	 Ya	 Ya	 Tidak
<u>AWS::CloudFront::Function</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::CloudFront::KeyGroup</u>	 Ya	 Ya	 Ya
<u>AWS::CloudFront::KeyValueStore</u>	 Ya	 Ya	 Tidak
<u>AWS::CloudFront::MonitoringSubscription</u>	 Ya	 Ya	 Tidak
<u>AWS::CloudFront::OriginAccessControl</u>	 Ya	 Ya	 Ya
<u>AWS::CloudFront::OriginRequestPolicy</u>	 Ya	 Ya	 Ya
<u>AWS::CloudFront::PublicKey</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::CloudFront::RealtimeLogConfig</u>	 Ya	 Ya	 Ya
<u>AWS::CloudFront::ResponseHeadersPolicy</u>	 Ya	 Ya	 Tidak
<u>AWS::CloudFront::VpcOrigin</u>	 Ya	 Ya	 Tidak
<u>AWS::CloudTrail::Channel</u>	 Ya	 Ya	 Tidak
<u>AWS::CloudTrail::Dashboard</u>	 Ya	 Ya	 Tidak
<u>AWS::CloudTrail::EventDataStore</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::CloudTrail::ResourcePolicy</u>	 Ya	 Ya	 Tidak
<u>AWS::CloudTrail::Trail</u>	 Ya	 Ya	 Ya
<u>AWS::CloudWatch::Alarm</u>	 Ya	 Ya	 Ya
<u>AWS::CloudWatch::CompositeAlarm</u>	 Ya	 Ya	 Ya
<u>AWS::CloudWatch::Dashboard</u>	 Ya	 Ya	 Tidak
<u>AWS::CloudWatch::MetricStream</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::CodeArtifact::Domain</u>	 Ya	 Ya	 Ya
<u>AWS::CodeArtifact::PackageGroup</u>	 Ya	 Ya	 Tidak
<u>AWS::CodeArtifact::Repository</u>	 Ya	 Ya	 Ya
<u>AWS::CodeBuild::Fleet</u>	 Ya	 Ya	 Tidak
<u>AWS::CodeConnections::Connection</u>	 Ya	 Ya	 Tidak
<u>AWS::CodeDeploy::Application</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::CodeDeploy::DeploymentConfig</u>	 Ya	 Ya	 Tidak
<u>AWS::CodeGuruProfiler::ProfilingGroup</u>	 Ya	 Ya	 Ya
<u>AWS::CodeGuruReviewer::RepositoryAssociation</u>	 Ya	 Ya	 Tidak
<u>AWS::CodePipeline::CustomActionType</u>	 Ya	 Ya	 Tidak
<u>AWS::CodePipeline::Pipeline</u>	 Ya	 Ya	 Tidak
<u>AWS::CodeStarConnections::Connection</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::CodeStarConnections::RepositoryLink</u>	 Ya	 Ya	 Tidak
<u>AWS::CodeStarConnections::SyncConfiguration</u>	 Ya	 Ya	 Tidak
<u>AWS::CodeStarNotifications::NotificationRule</u>	 Ya	 Ya	 Ya
<u>AWS::Cognito::IdentityPool</u>	 Ya	 Ya	 Tidak
<u>AWS::Cognito::IdentityPoolPrincipalTag</u>	 Ya	 Ya	 Ya
<u>AWS::Cognito::IdentityPoolRoleAttachment</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::Cognito::LogDeliveryConfiguration</u>	 Ya	 Tidak	 Tidak
<u>AWS::Cognito::ManagedLoginBranding</u>	 Ya	 Ya	 Tidak
<u>AWS::Cognito::UserPool</u>	 Ya	 Tidak	 Tidak
<u>AWS::Cognito::UserPoolClient</u>	 Ya	 Ya	 Tidak
<u>AWS::Cognito::UserPoolDomain</u>	 Ya	 Ya	 Tidak
<u>AWS::Cognito::UserPoolGroup</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::Cognito::UserPoolIdentityProvider</u>	 Ya	 Ya	 Tidak
<u>AWS::Cognito::UserPoolResourceServer</u>	 Ya	 Ya	 Tidak
<u>AWS::Cognito::UserPoolRiskConfigurationAttachment</u>	 Ya	 Ya	 Tidak
<u>AWS::Cognito::UserPoolUICustomizationAttachment</u>	 Ya	 Ya	 Tidak
<u>AWS::Cognito::UserPoolUser</u>	 Ya	 Tidak	 Tidak
<u>AWS::Cognito::UserPoolUserToGroupAttachment</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::Comprehend::DocumentClassifier</u>	 Ya	 Ya	 Ya
<u>AWS::Comprehend::Flywheel</u>	 Ya	 Ya	 Ya
<u>AWS::Config::AggregationAuthorization</u>	 Ya	 Ya	 Ya
<u>AWS::Config::ConfigRule</u>	 Ya	 Ya	 Ya
<u>AWS::Config::ConfigurationAggregator</u>	 Ya	 Ya	 Ya
<u>AWS::Config::ConformancePack</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::Config::OrganizationConformancePack</u>	 Ya	 Ya	 Ya
<u>AWS::Config::StoredQuery</u>	 Ya	 Ya	 Ya
<u>AWS::Connect::AgentStatus</u>	 Ya	 Ya	 Tidak
<u>AWS::Connect::ApprovedOrigin</u>	 Ya	 Ya	 Ya
<u>AWS::Connect::ContactFlow</u>	 Ya	 Ya	 Ya
<u>AWS::Connect::ContactFlowModule</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::Connect::ContactFlowVersion</u>	 Ya	 Ya	 Tidak
<u>AWS::Connect::EmailAddress</u>	 Ya	 Ya	 Tidak
<u>AWS::Connect::EvaluationForm</u>	 Ya	 Ya	 Ya
<u>AWS::Connect::HoursOfOperation</u>	 Ya	 Ya	 Tidak
<u>AWS::Connect::Instance</u>	 Ya	 Ya	 Ya
<u>AWS::Connect::InstanceStorageConfig</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::Connect::IntegrationAssociation</u>	 Ya	 Ya	 Ya
<u>AWS::Connect::PhoneNumber</u>	 Ya	 Ya	 Tidak
<u>AWS::Connect::PredefinedAttribute</u>	 Ya	 Ya	 Tidak
<u>AWS::Connect::Prompt</u>	 Ya	 Ya	 Ya
<u>AWS::Connect::Queue</u>	 Ya	 Ya	 Ya
<u>AWS::Connect::QuickConnect</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::Connect::RoutingProfile</u>	 Ya	 Ya	 Ya
<u>AWS::Connect::Rule</u>	 Ya	 Ya	 Tidak
<u>AWS::Connect::SecurityKey</u>	 Ya	 Ya	 Ya
<u>AWS::Connect::SecurityProfile</u>	 Ya	 Ya	 Ya
<u>AWS::Connect::TaskTemplate</u>	 Ya	 Ya	 Tidak
<u>AWS::Connect::TrafficDistributionGroup</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::Connect::User</u>	 Ya	 Ya	 Tidak
<u>AWS::Connect::UserHierarchyGroup</u>	 Ya	 Ya	 Tidak
<u>AWS::Connect::UserHierarchyStructure</u>	 Ya	 Ya	 Tidak
<u>AWS::Connect::View</u>	 Ya	 Ya	 Ya
<u>AWS::Connect::ViewVersion</u>	 Ya	 Ya	 Ya
<u>AWS::ConnectCampaigns::Campaign</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::ConnectCampaignsV2::Campaign</u>	 Ya	 Ya	 Tidak
<u>AWS::ControlTower::EnabledBaseline</u>	 Ya	 Ya	 Tidak
<u>AWS::ControlTower::EnabledControl</u>	 Ya	 Ya	 Ya
<u>AWS::ControlTower::LandingZone</u>	 Ya	 Ya	 Tidak
<u>AWS::CustomerProfiles::CalculatedAttributeDefinition</u>	 Ya	 Ya	 Ya
<u>AWS::CustomerProfiles::Domain</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::CustomerProfiles::EventStream</u>	 Ya	 Ya	 Ya
<u>AWS::CustomerProfiles::EventTrigger</u>	 Ya	 Ya	 Tidak
<u>AWS::CustomerProfiles::Integration</u>	 Ya	 Ya	 Ya
<u>AWS::CustomerProfiles::ObjectType</u>	 Ya	 Ya	 Ya
<u>AWS::CustomerProfiles::SegmentDefinition</u>	 Ya	 Ya	 Tidak
<u>AWS::DMS::DataMigration</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::DMS::DataProvider</u>	 Ya	 Ya	 Tidak
<u>AWS::DMS::InstanceProfile</u>	 Ya	 Ya	 Tidak
<u>AWS::DMS::MigrationProject</u>	 Ya	 Ya	 Tidak
<u>AWS::DMS::ReplicationConfig</u>	 Ya	 Ya	 Ya
<u>AWS::DSQL::Cluster</u>	 Ya	 Ya	 Tidak
<u>AWS::DataBrew::Dataset</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::DataBrew::Job</u>	 Ya	 Ya	 Ya
<u>AWS::DataBrew::Project</u>	 Ya	 Ya	 Ya
<u>AWS::DataBrew::Recipe</u>	 Ya	 Ya	 Ya
<u>AWS::DataBrew::Ruleset</u>	 Ya	 Ya	 Ya
<u>AWS::DataBrew::Schedule</u>	 Ya	 Ya	 Ya
<u>AWS::DataPipeline::Pipeline</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::DataSync::Agent</u>	 Ya	 Ya	 Ya
<u>AWS::DataSync::LocationAzureBlob</u>	 Ya	 Ya	 Ya
<u>AWS::DataSync::LocationEFS</u>	 Ya	 Ya	 Ya
<u>AWS::DataSync::LocationFSxLustre</u>	 Ya	 Ya	 Ya
<u>AWS::DataSync::LocationFSxONTAP</u>	 Ya	 Ya	 Ya
<u>AWS::DataSync::LocationFSxOpenZFS</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::DataSync::LocationFSxWindows</u>	 Ya	 Ya	 Ya
<u>AWS::DataSync::LocationHDFS</u>	 Ya	 Ya	 Ya
<u>AWS::DataSync::LocationNFS</u>	 Ya	 Ya	 Ya
<u>AWS::DataSync::LocationObjectStorage</u>	 Ya	 Ya	 Ya
<u>AWS::DataSync::LocationS3</u>	 Ya	 Ya	 Ya
<u>AWS::DataSync::LocationSMB</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::DataSync::Task</u>	 Ya	 Ya	 Ya
<u>AWS::DataZone::Connection</u>	 Ya	 Ya	 Tidak
<u>AWS::DataZone::DataSource</u>	 Ya	 Ya	 Tidak
<u>AWS::DataZone::Domain</u>	 Ya	 Ya	 Tidak
<u>AWS::DataZone::DomainUnit</u>	 Ya	 Ya	 Tidak
<u>AWS::DataZone::Environment</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::DataZone::EnvironmentActions</u>	 Ya	 Ya	 Tidak
<u>AWS::DataZone::EnvironmentBlueprintConfiguration</u>	 Ya	 Ya	 Tidak
<u>AWS::DataZone::EnvironmentProfile</u>	 Ya	 Ya	 Tidak
<u>AWS::DataZone::GroupProfile</u>	 Ya	 Ya	 Tidak
<u>AWS::DataZone::Owner</u>	 Ya	 Ya	 Tidak
<u>AWS::DataZone::Project</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::DataZone::ProjectMembership</u>	 Ya	 Ya	 Tidak
<u>AWS::DataZone::ProjectProfile</u>	 Ya	 Ya	 Tidak
<u>AWS::DataZone::SubscriptionTarget</u>	 Ya	 Ya	 Tidak
<u>AWS::DataZone::UserProfile</u>	 Ya	 Ya	 Tidak
<u>AWS::Deadline::Farm</u>	 Ya	 Ya	 Tidak
<u>AWS::Deadline::Fleet</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::Deadline::LicenseEndpoint</u>	 Ya	 Ya	 Tidak
<u>AWS::Deadline::Limit</u>	 Ya	 Ya	 Tidak
<u>AWS::Deadline::MeteredProduct</u>	 Ya	 Ya	 Tidak
<u>AWS::Deadline::Monitor</u>	 Ya	 Ya	 Tidak
<u>AWS::Deadline::Queue</u>	 Ya	 Ya	 Tidak
<u>AWS::Deadline::QueueEnvironment</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::Deadline::QueueFleetAssociation</u>	 Ya	 Ya	 Tidak
<u>AWS::Deadline::QueueLimitAssociation</u>	 Ya	 Ya	 Tidak
<u>AWS::Deadline::StorageProfile</u>	 Ya	 Ya	 Tidak
<u>AWS::Detective::Graph</u>	 Ya	 Ya	 Ya
<u>AWS::Detective::MemberInvitation</u>	 Ya	 Ya	 Ya
<u>AWS::Detective::OrganizationAdmin</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::DevOpsGuru::LogAnomalyDetectionIntegration</u>	 Ya	 Ya	 Ya
<u>AWS::DevOpsGuru::NotificationChannel</u>	 Ya	 Ya	 Ya
<u>AWS::DevOpsGuru::ResourceCollection</u>	 Ya	 Ya	 Ya
<u>AWS::DeviceFarm::DevicePool</u>	 Ya	 Ya	 Tidak
<u>AWS::DeviceFarm::InstanceProfile</u>	 Ya	 Ya	 Tidak
<u>AWS::DeviceFarm::NetworkProfile</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::DeviceFarm::Project</u>	 Ya	 Ya	 Tidak
<u>AWS::DeviceFarm::TestGridProject</u>	 Ya	 Ya	 Tidak
<u>AWS::DeviceFarm::VPCEConfiguration</u>	 Ya	 Ya	 Tidak
<u>AWS::DirectoryService::SimpleAD</u>	 Ya	 Ya	 Ya
<u>AWS::DocDBElastic::Cluster</u>	 Ya	 Ya	 Ya
<u>AWS::DynamoDB::GlobalTable</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::DynamoDB::Table</u>	 Ya	 Ya	 Ya
<u>AWS::EC2::CapacityReservation</u>	 Ya	 Ya	 Ya
<u>AWS::EC2::CapacityReservationFleet</u>	 Ya	 Ya	 Ya
<u>AWS::EC2::CarrierGateway</u>	 Ya	 Ya	 Ya
<u>AWS::EC2::CustomerGateway</u>	 Ya	 Ya	 Ya
<u>AWS::EC2::DHCPOptions</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::EC2::EC2Fleet</u>	 Ya	 Ya	 Tidak
<u>AWS::EC2::EIP</u>	 Ya	 Ya	 Ya
<u>AWS::EC2::EIPAssociation</u>	 Ya	 Tidak	 Ya
<u>AWS::EC2::EgressOnlyInternetGateway</u>	 Ya	 Ya	 Ya
<u>AWS::EC2::EnclaveCertificateIamRoleAssociation</u>	 Ya	 Ya	 Ya
<u>AWS::EC2::FlowLog</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::EC2::GatewayRouteTableAssociation</u>	 Ya	 Ya	 Tidak
<u>AWS::EC2::Host</u>	 Ya	 Ya	 Ya
<u>AWS::EC2::IPAM</u>	 Ya	 Ya	 Ya
<u>AWS::EC2::IPAMAllocation</u>	 Ya	 Ya	 Ya
<u>AWS::EC2::IPAMPool</u>	 Ya	 Ya	 Ya
<u>AWS::EC2::IPAMPoolCidr</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::EC2::IPAMResourceDiscovery</u>	 Ya	 Ya	 Ya
<u>AWS::EC2::IPAMResourceDiscoveryAssociation</u>	 Ya	 Ya	 Ya
<u>AWS::EC2::IPAMScope</u>	 Ya	 Ya	 Ya
<u>AWS::EC2::Instance</u>	 Ya	 Ya	 Ya
<u>AWS::EC2::InstanceConnectEndpoint</u>	 Ya	 Ya	 Ya
<u>AWS::EC2::InternetGateway</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAC
<u>AWS::EC2::KeyPair</u>	 Ya	 Ya	 Ya
<u>AWS::EC2::LaunchTemplate</u>	 Ya	 Ya	 Tidak
<u>AWS::EC2::LocalGatewayRoute</u>	 Ya	 Ya	 Ya
<u>AWS::EC2::LocalGatewayRouteTable</u>	 Ya	 Ya	 Ya
<u>AWS::EC2::LocalGatewayRouteTableVPCAssociation</u>	 Ya	 Ya	 Ya
<u>AWS::EC2::LocalGatewayRouteTableVirtualInterfaceGroupAssociation</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::EC2::NatGateway</u>	 Ya	 Ya	 Ya
<u>AWS::EC2::NetworkAcl</u>	 Ya	 Ya	 Ya
<u>AWS::EC2::NetworkInsightsAccessScope</u>	 Ya	 Ya	 Tidak
<u>AWS::EC2::NetworkInsightsAccessScope Analysis</u>	 Ya	 Ya	 Ya
<u>AWS::EC2::NetworkInsightsAnalysis</u>	 Ya	 Ya	 Tidak
<u>AWS::EC2::NetworkInsightsPath</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::EC2::NetworkInterface</u>	 Ya	 Ya	 Ya
<u>AWS::EC2::NetworkInterfaceAttachment</u>	 Ya	 Ya	 Ya
<u>AWS::EC2::NetworkPerformanceMetricSubscription</u>	 Ya	 Ya	 Ya
<u>AWS::EC2::PlacementGroup</u>	 Ya	 Ya	 Ya
<u>AWS::EC2::PrefixList</u>	 Ya	 Ya	 Tidak
<u>AWS::EC2::Route</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::EC2::RouteServer</u>	 Ya	 Ya	 Tidak
<u>AWS::EC2::RouteServerAssociation</u>	 Ya	 Ya	 Tidak
<u>AWS::EC2::RouteServerEndpoint</u>	 Ya	 Ya	 Tidak
<u>AWS::EC2::RouteServerPeer</u>	 Ya	 Ya	 Tidak
<u>AWS::EC2::RouteServerPropagation</u>	 Ya	 Ya	 Tidak
<u>AWS::EC2::RouteTable</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::EC2::SecurityGroup</u>	 Ya	 Ya	 Ya
<u>AWS::EC2::SecurityGroupEgress</u>	 Ya	 Ya	 Tidak
<u>AWS::EC2::SecurityGroupIngress</u>	 Ya	 Ya	 Tidak
<u>AWS::EC2::SecurityGroupVpcAssociation</u>	 Ya	 Ya	 Tidak
<u>AWS::EC2::SnapshotBlockPublicAccess</u>	 Ya	 Ya	 Tidak
<u>AWS::EC2::SpotFleet</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::EC2::Subnet</u>	 Ya	 Ya	 Ya
<u>AWS::EC2::SubnetCidrBlock</u>	 Ya	 Ya	 Ya
<u>AWS::EC2::SubnetNetworkAclAssociation</u>	 Ya	 Ya	 Ya
<u>AWS::EC2::SubnetRouteTableAssociation</u>	 Ya	 Tidak	 Ya
<u>AWS::EC2::TrafficMirrorFilter</u>	 Ya	 Ya	 Tidak
<u>AWS::EC2::TrafficMirrorFilterRule</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::EC2::TransitGateway</u>	 Ya	 Ya	 Ya
<u>AWS::EC2::TransitGatewayAttachment</u>	 Ya	 Ya	 Ya
<u>AWS::EC2::TransitGatewayConnect</u>	 Ya	 Ya	 Ya
<u>AWS::EC2::TransitGatewayMulticastDomain</u>	 Ya	 Ya	 Ya
<u>AWS::EC2::TransitGatewayMulticastDomainAssociation</u>	 Ya	 Ya	 Ya
<u>AWS::EC2::TransitGatewayMulticastGroupMember</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
AWS::EC2::TransitGatewayMulticastGroupSource	 Ya	 Ya	 Ya
AWS::EC2::TransitGatewayPeeringAttachment	 Ya	 Ya	 Ya
AWS::EC2::TransitGatewayRoute	 Ya	 Ya	 Tidak
AWS::EC2::TransitGatewayRouteTable	 Ya	 Ya	 Ya
AWS::EC2::TransitGatewayRouteTableAssociation	 Ya	 Ya	 Tidak
AWS::EC2::TransitGatewayRouteTablePropagation	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
AWS::EC2::TransitGatewayVpcAttachment	 Ya	 Ya	 Ya
AWS::EC2::VPC	 Ya	 Ya	 Ya
AWS::EC2::VPCLockPublicAccessExclusion	 Ya	 Ya	 Tidak
AWS::EC2::VPCLockPublicAccessOptions	 Ya	 Ya	 Tidak
AWS::EC2::VPCCIDRBlock	 Ya	 Ya	 Tidak
AWS::EC2::VPCHDCPOptionsAssociation	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
AWS::EC2::VPCEndpoint	 Ya	 Ya	 Ya
AWS::EC2::VPCEndpointConnectionNotification	 Ya	 Ya	 Ya
AWS::EC2::VPCEndpointService	 Ya	 Ya	 Ya
AWS::EC2::VPCEndpointServicePermissions	 Ya	 Ya	 Tidak
AWS::EC2::VPCGatewayAttachment	 Ya	 Ya	 Tidak
AWS::EC2::VPCPeeringConnection	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::EC2::VPNConnection</u>	 Ya	 Ya	 Ya
<u>AWS::EC2::VPNConnectionRoute</u>	 Ya	 Ya	 Ya
<u>AWS::EC2::VPNGateway</u>	 Ya	 Ya	 Ya
<u>AWS::EC2::VerifiedAccessEndpoint</u>	 Ya	 Ya	 Ya
<u>AWS::EC2::VerifiedAccessGroup</u>	 Ya	 Ya	 Ya
<u>AWS::EC2::VerifiedAccessInstance</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::EC2::VerifiedAccessTrustProvider</u>	 Ya	 Ya	 Ya
<u>AWS::EC2::Volume</u>	 Ya	 Ya	 Ya
<u>AWS::EC2::VolumeAttachment</u>	 Ya	 Ya	 Ya
<u>AWS::ECR::PullThroughCacheRule</u>	 Ya	 Ya	 Ya
<u>AWS::ECR::RegistryPolicy</u>	 Ya	 Ya	 Ya
<u>AWS::ECR::RegistryScanningConfiguration</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::ECR::ReplicationConfiguration</u>	 Ya	 Ya	 Ya
<u>AWS::ECR::Repository</u>	 Ya	 Ya	 Ya
<u>AWS::ECR::RepositoryCreationTemplate</u>	 Ya	 Ya	 Tidak
<u>AWS::ECS::CapacityProvider</u>	 Ya	 Ya	 Tidak
<u>AWS::ECS::Cluster</u>	 Ya	 Tidak	 Ya
<u>AWS::ECS::ClusterCapacityProviderAssociations</u>	 Ya	 Tidak	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::ECS::PrimaryTaskSet</u>	 Ya	 Ya	 Tidak
<u>AWS::ECS::Service</u>	 Ya	 Ya	 Ya
<u>AWS::ECS::TaskDefinition</u>	 Ya	 Ya	 Ya
<u>AWS::ECS::TaskSet</u>	 Ya	 Ya	 Tidak
<u>AWS::EFS::AccessPoint</u>	 Ya	 Ya	 Ya
<u>AWS::EFS::FileSystem</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::EFS::MountTarget</u>	 Ya	 Ya	 Ya
<u>AWS::EKS::AccessEntry</u>	 Ya	 Ya	 Tidak
<u>AWS::EKS::Addon</u>	 Ya	 Ya	 Ya
<u>AWS::EKS::Cluster</u>	 Ya	 Tidak	 Ya
<u>AWS::EKS::FargateProfile</u>	 Ya	 Ya	 Ya
<u>AWS::EKS::IdentityProviderConfig</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::EKS::Nodegroup</u>	 Ya	 Tidak	 Ya
<u>AWS::EKS::PodIdentityAssociation</u>	 Ya	 Ya	 Tidak
<u>AWS::EMR::SecurityConfiguration</u>	 Ya	 Ya	 Ya
<u>AWS::EMR::Step</u>	 Ya	 Tidak	 Tidak
<u>AWS::EMR::Studio</u>	 Ya	 Ya	 Ya
<u>AWS::EMR::StudioSessionMapping</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::EMR::WALWorkspace</u>	 Ya	 Ya	 Tidak
<u>AWS::EMRContainers::VirtualCluster</u>	 Ya	 Ya	 Tidak
<u>AWS::EMRServerless::Application</u>	 Ya	 Ya	 Ya
<u>AWS::EVS::Environment</u>	 Ya	 Ya	 Tidak
<u>AWS::ElastiCache::GlobalReplicationGroup</u>	 Ya	 Ya	 Tidak
<u>AWS::ElastiCache::ParameterGroup</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::ElastiCache::ServerlessCache</u>	 Ya	 Ya	 Tidak
<u>AWS::ElastiCache::SubnetGroup</u>	 Ya	 Ya	 Ya
<u>AWS::ElastiCache::User</u>	 Ya	 Ya	 Ya
<u>AWS::ElastiCache::UserGroup</u>	 Ya	 Ya	 Ya
<u>AWS::ElasticBeanstalk::Application</u>	 Ya	 Ya	 Ya
<u>AWS::ElasticBeanstalk::ApplicationVersion</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAC
<u>AWS::ElasticBeanstalk::ConfigurationTemplate</u>	 Ya	 Ya	 Ya
<u>AWS::ElasticBeanstalk::Environment</u>	 Ya	 Tidak	 Tidak
<u>AWS::ElasticLoadBalancingV2::Listener</u>	 Ya	 Ya	 Ya
<u>AWS::ElasticLoadBalancingV2::ListenerRule</u>	 Ya	 Tidak	 Ya
<u>AWS::ElasticLoadBalancingV2::LoadBalancer</u>	 Ya	 Ya	 Ya
<u>AWS::ElasticLoadBalancingV2::TargetGroup</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::ElasticLoadBalancingV2::TrustStore</u>	 Ya	 Ya	 Tidak
<u>AWS::ElasticLoadBalancingV2::TrustStoreRevocation</u>	 Ya	 Ya	 Tidak
<u>AWS::EntityResolution::IdMappingWorkflow</u>	 Ya	 Ya	 Ya
<u>AWS::EntityResolution::IdNamespace</u>	 Ya	 Ya	 Tidak
<u>AWS::EntityResolution::MatchingWorkflow</u>	 Ya	 Ya	 Ya
<u>AWS::EntityResolution::PolicyStatement</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::EntityResolution::SchemaMapping</u>	 Ya	 Ya	 Ya
<u>AWS::EventSchemas::Discoverer</u>	 Ya	 Ya	 Tidak
<u>AWS::EventSchemas::Registry</u>	 Ya	 Ya	 Tidak
<u>AWS::EventSchemas::RegistryPolicy</u>	 Ya	 Ya	 Tidak
<u>AWS::EventSchemas::Schema</u>	 Ya	 Ya	 Tidak
<u>AWS::Events::ApiDestination</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::Events::Archive</u>	 Ya	 Ya	 Ya
<u>AWS::Events::Connection</u>	 Ya	 Ya	 Tidak
<u>AWS::Events::Endpoint</u>	 Ya	 Ya	 Ya
<u>AWS::Events::EventBus</u>	 Ya	 Ya	 Ya
<u>AWS::Events::Rule</u>	 Ya	 Ya	 Ya
<u>AWS::Evidently::Experiment</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::Evidently::Feature</u>	 Ya	 Tidak	 Tidak
<u>AWS::Evidently::Launch</u>	 Ya	 Ya	 Tidak
<u>AWS::Evidently::Project</u>	 Ya	 Ya	 Tidak
<u>AWS::Evidently::Segment</u>	 Ya	 Ya	 Tidak
<u>AWS::FIS::ExperimentTemplate</u>	 Ya	 Ya	 Ya
<u>AWS::FIS::TargetAccountConfiguration</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::FMS::NotificationChannel</u>	 Ya	 Ya	 Tidak
<u>AWS::FMS::Policy</u>	 Ya	 Tidak	 Tidak
<u>AWS::FMS::ResourceSet</u>	 Ya	 Ya	 Tidak
<u>AWS::FSx::DataRepositoryAssociation</u>	 Ya	 Ya	 Ya
<u>AWS::FSx::S3AccessPointAttachment</u>	 Ya	 Ya	 Tidak
<u>AWS::FinSpace::Environment</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::Forecast::Dataset</u>	 Ya	 Ya	 Tidak
<u>AWS::Forecast::DatasetGroup</u>	 Ya	 Ya	 Tidak
<u>AWS::FraudDetector::Detector</u>	 Ya	 Ya	 Ya
<u>AWS::FraudDetector::EntityType</u>	 Ya	 Ya	 Ya
<u>AWS::FraudDetector::EventType</u>	 Ya	 Ya	 Ya
<u>AWS::FraudDetector::Label</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::FraudDetector::List</u>	 Ya	 Ya	 Ya
<u>AWS::FraudDetector::Outcome</u>	 Ya	 Ya	 Ya
<u>AWS::FraudDetector::Variable</u>	 Ya	 Ya	 Ya
<u>AWS::GameLift::Alias</u>	 Ya	 Ya	 Ya
<u>AWS::GameLift::Build</u>	 Ya	 Ya	 Ya
<u>AWS::GameLift::ContainerFleet</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::GameLift::ContainerGroupDefinition</u>	 Ya	 Ya	 Tidak
<u>AWS::GameLift::Fleet</u>	 Ya	 Ya	 Ya
<u>AWS::GameLift::GameServerGroup</u>	 Ya	 Ya	 Tidak
<u>AWS::GameLift::GameSessionQueue</u>	 Ya	 Ya	 Tidak
<u>AWS::GameLift::Location</u>	 Ya	 Ya	 Tidak
<u>AWS::GameLift::MatchmakingConfiguration</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::GameLift::MatchmakingRuleSet</u>	 Ya	 Ya	 Tidak
<u>AWS::GameLift::Script</u>	 Ya	 Ya	 Tidak
<u>AWS::GameLiftStreams::Application</u>	 Ya	 Ya	 Tidak
<u>AWS::GameLiftStreams::StreamGroup</u>	 Ya	 Ya	 Tidak
<u>AWS::GlobalAccelerator::Accelerator</u>	 Ya	 Ya	 Ya
<u>AWS::GlobalAccelerator::CrossAccount Attachment</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::GlobalAccelerator::EndpointGroup</u>	 Ya	 Ya	 Ya
<u>AWS::GlobalAccelerator::Listener</u>	 Ya	 Ya	 Ya
<u>AWS::Glue::Crawler</u>	 Ya	 Ya	 Tidak
<u>AWS::Glue::Database</u>	 Ya	 Ya	 Tidak
<u>AWS::Glue::Job</u>	 Ya	 Ya	 Tidak
<u>AWS::Glue::Registry</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::Glue::Schema</u>	 Ya	 Ya	 Ya
<u>AWS::Glue::SchemaVersion</u>	 Ya	 Ya	 Ya
<u>AWS::Glue::SchemaVersionMetadata</u>	 Ya	 Ya	 Ya
<u>AWS::Glue::Trigger</u>	 Ya	 Ya	 Tidak
<u>AWS::Glue::UsageProfile</u>	 Ya	 Ya	 Tidak
<u>AWS::Grafana::Workspace</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::GreengrassV2::ComponentVersion</u>	 Ya	 Ya	 Tidak
<u>AWS::GreengrassV2::Deployment</u>	 Ya	 Ya	 Ya
<u>AWS::GroundStation::Config</u>	 Ya	 Ya	 Ya
<u>AWS::GroundStation::DataflowEndpoint Group</u>	 Ya	 Ya	 Ya
<u>AWS::GroundStation::MissionProfile</u>	 Ya	 Ya	 Ya
<u>AWS::GuardDuty::Detector</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAC
<u>AWS::GuardDuty::Filter</u>	 Ya	 Ya	 Tidak
<u>AWS::GuardDuty::IPSet</u>	 Ya	 Ya	 Tidak
<u>AWS::GuardDuty::MalwareProtectionPlan</u>	 Ya	 Ya	 Tidak
<u>AWS::GuardDuty::Master</u>	 Ya	 Ya	 Tidak
<u>AWS::GuardDuty::Member</u>	 Ya	 Ya	 Tidak
<u>AWS::GuardDuty::PublishingDestination</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAC
<u>AWS::GuardDuty::ThreatIntelSet</u>	 Ya	 Ya	 Tidak
<u>AWS::HealthImaging::Datastore</u>	 Ya	 Ya	 Ya
<u>AWS::HealthLake::FHIRDatastore</u>	 Ya	 Ya	 Ya
<u>AWS::IAM::Group</u>	 Ya	 Ya	 Ya
<u>AWS::IAM::GroupPolicy</u>	 Ya	 Ya	 Tidak
<u>AWS::IAM::InstanceProfile</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::IAM::ManagedPolicy</u>	 Ya	 Ya	 Ya
<u>AWS::IAM::OIDCProvider</u>	 Ya	 Ya	 Ya
<u>AWS::IAM::Role</u>	 Ya	 Ya	 Ya
<u>AWS::IAM::RolePolicy</u>	 Ya	 Ya	 Tidak
<u>AWS::IAM::SAMLProvider</u>	 Ya	 Ya	 Ya
<u>AWS::IAM::ServerCertificate</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::IAM::ServiceLinkedRole</u>	 Ya	 Ya	 Tidak
<u>AWS::IAM::User</u>	 Ya	 Ya	 Ya
<u>AWS::IAM::UserPolicy</u>	 Ya	 Ya	 Tidak
<u>AWS::IAM::VirtualMFADevice</u>	 Ya	 Ya	 Tidak
<u>AWS::IVS::Channel</u>	 Ya	 Ya	 Ya
<u>AWS::IVS::EncoderConfiguration</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::IVS::IngestConfiguration</u>	 Ya	 Ya	 Tidak
<u>AWS::IVS::PlaybackKeyPair</u>	 Ya	 Ya	 Ya
<u>AWS::IVS::PlaybackRestrictionPolicy</u>	 Ya	 Ya	 Tidak
<u>AWS::IVS::PublicKey</u>	 Ya	 Ya	 Tidak
<u>AWS::IVS::RecordingConfiguration</u>	 Ya	 Ya	 Ya
<u>AWS::IVS::Stage</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::IVS::StorageConfiguration</u>	 Ya	 Ya	 Tidak
<u>AWS::IVS::StreamKey</u>	 Ya	 Ya	 Tidak
<u>AWS::IVSChat::LoggingConfiguration</u>	 Ya	 Ya	 Ya
<u>AWS::IVSChat::Room</u>	 Ya	 Ya	 Ya
<u>AWS::IdentityStore::Group</u>	 Ya	 Ya	 Ya
<u>AWS::IdentityStore::GroupMembership</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::ImageBuilder::Component</u>	 Ya	 Ya	 Ya
<u>AWS::ImageBuilder::ContainerRecipe</u>	 Ya	 Ya	 Tidak
<u>AWS::ImageBuilder::DistributionConfiguration</u>	 Ya	 Ya	 Ya
<u>AWS::ImageBuilder::Image</u>	 Ya	 Ya	 Ya
<u>AWS::ImageBuilder::ImagePipeline</u>	 Ya	 Ya	 Ya
<u>AWS::ImageBuilder::ImageRecipe</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::ImageBuilder::InfrastructureConfiguration</u>	 Ya	 Ya	 Ya
<u>AWS::ImageBuilder::LifecyclePolicy</u>	 Ya	 Ya	 Tidak
<u>AWS::ImageBuilder::Workflow</u>	 Ya	 Ya	 Tidak
<u>AWS::Inspector::AssessmentTarget</u>	 Ya	 Ya	 Ya
<u>AWS::Inspector::AssessmentTemplate</u>	 Ya	 Ya	 Ya
<u>AWS::Inspector::ResourceGroup</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::InspectorV2::CisScanConfiguration</u>	 Ya	 Ya	 Tidak
<u>AWS::InspectorV2::Filter</u>	 Ya	 Ya	 Tidak
<u>AWS::InternetMonitor::Monitor</u>	 Ya	 Ya	 Ya
<u>AWS::Invoicing::InvoiceUnit</u>	 Ya	 Ya	 Tidak
<u>AWS::IoT::AccountAuditConfiguration</u>	 Ya	 Ya	 Ya
<u>AWS::IoT::Authorizer</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::IoT::BillingGroup</u>	 Ya	 Ya	 Ya
<u>AWS::IoT::CACertificate</u>	 Ya	 Ya	 Ya
<u>AWS::IoT::Certificate</u>	 Ya	 Ya	 Ya
<u>AWS::IoT::CertificateProvider</u>	 Ya	 Ya	 Tidak
<u>AWS::IoT::Command</u>	 Ya	 Ya	 Tidak
<u>AWS::IoT::CustomMetric</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::IoT::Dimension</u>	 Ya	 Ya	 Ya
<u>AWS::IoT::DomainConfiguration</u>	 Ya	 Ya	 Tidak
<u>AWS::IoT::FleetMetric</u>	 Ya	 Ya	 Tidak
<u>AWS::IoT::JobTemplate</u>	 Ya	 Ya	 Tidak
<u>AWS::IoT::Logging</u>	 Ya	 Ya	 Ya
<u>AWS::IoT::MitigationAction</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::IoT::Policy</u>	 Ya	 Ya	 Ya
<u>AWS::IoT::ProvisioningTemplate</u>	 Ya	 Ya	 Ya
<u>AWS::IoT::ResourceSpecificLogging</u>	 Ya	 Ya	 Ya
<u>AWS::IoT::RoleAlias</u>	 Ya	 Ya	 Ya
<u>AWS::IoT::ScheduledAudit</u>	 Ya	 Ya	 Ya
<u>AWS::IoT::SecurityProfile</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::IoT::SoftwarePackage</u>	 Ya	 Ya	 Ya
<u>AWS::IoT::SoftwarePackageVersion</u>	 Ya	 Ya	 Tidak
<u>AWS::IoT::Thing</u>	 Ya	 Ya	 Ya
<u>AWS::IoT::ThingGroup</u>	 Ya	 Ya	 Ya
<u>AWS::IoT::ThingType</u>	 Ya	 Ya	 Tidak
<u>AWS::IoT::TopicRule</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::IoT::TopicRuleDestination</u>	 Ya	 Ya	 Ya
<u>AWS::IoTAnalytics::Channel</u>	 Ya	 Ya	 Ya
<u>AWS::IoTAnalytics::Dataset</u>	 Ya	 Ya	 Ya
<u>AWS::IoTAnalytics::Datastore</u>	 Ya	 Ya	 Ya
<u>AWS::IoTAnalytics::Pipeline</u>	 Ya	 Ya	 Ya
<u>AWS::IoTCoreDeviceAdvisor::SuiteDefinition</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::IoTEvents::AlarmModel</u>	 Ya	 Ya	 Ya
<u>AWS::IoTEvents::DetectorModel</u>	 Ya	 Ya	 Ya
<u>AWS::IoTEvents::Input</u>	 Ya	 Ya	 Ya
<u>AWS::IoTFleetHub::Application</u>	 Ya	 Ya	 Ya
<u>AWS::IoTSiteWise::AccessPolicy</u>	 Ya	 Ya	 Tidak
<u>AWS::IoTSiteWise::Asset</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::IoTSiteWise::AssetModel</u>	 Ya	 Ya	 Ya
<u>AWS::IoTSiteWise::Dashboard</u>	 Ya	 Ya	 Tidak
<u>AWS::IoTSiteWise::Dataset</u>	 Ya	 Ya	 Tidak
<u>AWS::IoTSiteWise::Gateway</u>	 Ya	 Ya	 Ya
<u>AWS::IoTSiteWise::Portal</u>	 Ya	 Ya	 Tidak
<u>AWS::IoTSiteWise::Project</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::IoTTwinMaker::ComponentType</u>	 Ya	 Ya	 Ya
<u>AWS::IoTTwinMaker::Entity</u>	 Ya	 Ya	 Ya
<u>AWS::IoTTwinMaker::Scene</u>	 Ya	 Ya	 Ya
<u>AWS::IoTTwinMaker::SyncJob</u>	 Ya	 Ya	 Ya
<u>AWS::IoTTwinMaker::Workspace</u>	 Ya	 Ya	 Ya
<u>AWS::IoTWireless::Destination</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAC
<u>AWS::IoTWireless::DeviceProfile</u>	 Ya	 Ya	 Ya
<u>AWS::IoTWireless::FuotaTask</u>	 Ya	 Ya	 Ya
<u>AWS::IoTWireless::MulticastGroup</u>	 Ya	 Ya	 Ya
<u>AWS::IoTWireless::NetworkAnalyzerConfiguration</u>	 Ya	 Ya	 Ya
<u>AWS::IoTWireless::ServiceProfile</u>	 Ya	 Ya	 Ya
<u>AWS::IoTWireless::TaskDefinition</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::IoTWireless::WirelessDevice</u>	 Ya	 Ya	 Ya
<u>AWS::IoTWireless::WirelessGateway</u>	 Ya	 Ya	 Ya
<u>AWS::KMS::Alias</u>	 Ya	 Tidak	 Ya
<u>AWS::KMS::Key</u>	 Ya	 Tidak	 Ya
<u>AWS::KMS::ReplicaKey</u>	 Ya	 Tidak	 Ya
<u>AWS::KafkaConnect::Connector</u>	 Ya	 Tidak	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::KafkaConnect::CustomPlugin</u>	 Ya	 Ya	 Tidak
<u>AWS::KafkaConnect::WorkerConfiguration</u>	 Ya	 Ya	 Tidak
<u>AWS::Kendra::DataSource</u>	 Ya	 Ya	 Ya
<u>AWS::Kendra::FAQ</u>	 Ya	 Ya	 Ya
<u>AWS::Kendra::Index</u>	 Ya	 Ya	 Ya
<u>AWS::KendraRanking::ExecutionPlan</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::Kinesis::ResourcePolicy</u>	 Ya	 Ya	 Tidak
<u>AWS::Kinesis::Stream</u>	 Ya	 Ya	 Ya
<u>AWS::KinesisAnalyticsV2::Application</u>	 Ya	 Ya	 Tidak
<u>AWS::KinesisFirehose::DeliveryStream</u>	 Ya	 Tidak	 Tidak
<u>AWS::KinesisVideo::SignalingChannel</u>	 Ya	 Ya	 Tidak
<u>AWS::KinesisVideo::Stream</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::LakeFormation::DataCellsFilter</u>	 Ya	 Ya	 Tidak
<u>AWS::LakeFormation::PrincipalPermissions</u>	 Ya	 Ya	 Tidak
<u>AWS::LakeFormation::Tag</u>	 Ya	 Ya	 Tidak
<u>AWS::LakeFormation::TagAssociation</u>	 Ya	 Ya	 Tidak
<u>AWS::Lambda::Alias</u>	 Ya	 Ya	 Tidak
<u>AWS::Lambda::CodeSigningConfig</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAC
<u>AWS::Lambda::EventInvokeConfig</u>	 Ya	 Ya	 Tidak
<u>AWS::Lambda::EventSourceMapping</u>	 Ya	 Ya	 Ya
<u>AWS::Lambda::Function</u>	 Ya	 Ya	 Ya
<u>AWS::Lambda::LayerVersion</u>	 Ya	 Ya	 Tidak
<u>AWS::Lambda::LayerVersionPermission</u>	 Ya	 Ya	 Ya
<u>AWS::Lambda::Permission</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
AWS::Lambda::Url	 Ya	 Ya	 Ya
AWS::Lambda::Version	 Ya	 Ya	 Ya
AWS::LaunchWizard::Deployment	 Ya	 Ya	 Tidak
AWS::Lex::Bot	 Ya	 Ya	 Tidak
AWS::Lex::BotAlias	 Ya	 Ya	 Tidak
AWS::Lex::BotVersion	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::Lex::ResourcePolicy</u>	 Ya	 Ya	 Tidak
<u>AWS::LicenseManager::Grant</u>	 Ya	 Ya	 Tidak
<u>AWS::LicenseManager::License</u>	 Ya	 Ya	 Tidak
<u>AWS::Lightsail::Alarm</u>	 Ya	 Ya	 Ya
<u>AWS::Lightsail::Bucket</u>	 Ya	 Ya	 Ya
<u>AWS::Lightsail::Certificate</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::Lightsail::Container</u>	 Ya	 Ya	 Ya
<u>AWS::Lightsail::Database</u>	 Ya	 Ya	 Tidak
<u>AWS::Lightsail::Disk</u>	 Ya	 Ya	 Ya
<u>AWS::Lightsail::Instance</u>	 Ya	 Ya	 Ya
<u>AWS::Lightsail::InstanceSnapshot</u>	 Ya	 Ya	 Tidak
<u>AWS::Lightsail::LoadBalancer</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::Lightsail::LoadBalancerTlsCertificate</u>	 Ya	 Ya	 Ya
<u>AWS::Lightsail::StaticIp</u>	 Ya	 Ya	 Ya
<u>AWS::Location::APIKey</u>	 Ya	 Ya	 Tidak
<u>AWS::Location::GeofenceCollection</u>	 Ya	 Ya	 Ya
<u>AWS::Location::Map</u>	 Ya	 Ya	 Ya
<u>AWS::Location::PlaceIndex</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::Location::RouteCalculator</u>	 Ya	 Ya	 Ya
<u>AWS::Location::Tracker</u>	 Ya	 Ya	 Ya
<u>AWS::Location::TrackerConsumer</u>	 Ya	 Ya	 Ya
<u>AWS::Logs::AccountPolicy</u>	 Ya	 Ya	 Ya
<u>AWS::Logs::Delivery</u>	 Ya	 Ya	 Tidak
<u>AWS::Logs::DeliveryDestination</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::Logs::DeliverySource</u>	 Ya	 Ya	 Tidak
<u>AWS::Logs::Destination</u>	 Ya	 Ya	 Ya
<u>AWS::Logs::Integration</u>	 Ya	 Ya	 Tidak
<u>AWS::Logs::LogAnomalyDetector</u>	 Ya	 Ya	 Tidak
<u>AWS::Logs::LogGroup</u>	 Ya	 Ya	 Ya
<u>AWS::Logs::LogStream</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::Logs::MetricFilter</u>	 Ya	 Ya	 Ya
<u>AWS::Logs::QueryDefinition</u>	 Ya	 Ya	 Tidak
<u>AWS::Logs::ResourcePolicy</u>	 Ya	 Ya	 Ya
<u>AWS::Logs::SubscriptionFilter</u>	 Ya	 Ya	 Ya
<u>AWS::Logs::Transformer</u>	 Ya	 Ya	 Tidak
<u>AWS::LookoutMetrics::Alert</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::LookoutMetrics::AnomalyDetector</u>	 Ya	 Ya	 Tidak
<u>AWS::LookoutVision::Project</u>	 Ya	 Ya	 Ya
<u>AWS::M2::Application</u>	 Ya	 Ya	 Tidak
<u>AWS::M2::Deployment</u>	 Ya	 Ya	 Tidak
<u>AWS::M2::Environment</u>	 Ya	 Ya	 Ya
<u>AWS::MSK::BatchScramSecret</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::MSK::Cluster</u>	 Ya	 Ya	 Ya
<u>AWS::MSK::ClusterPolicy</u>	 Ya	 Ya	 Ya
<u>AWS::MSK::Configuration</u>	 Ya	 Ya	 Ya
<u>AWS::MSK::Replicator</u>	 Ya	 Ya	 Ya
<u>AWS::MSK::ServerlessCluster</u>	 Ya	 Ya	 Ya
<u>AWS::MSK::VpcConnection</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::MWAA::Environment</u>	 Ya	 Ya	 Tidak
<u>AWS::Macie::AllowList</u>	 Ya	 Ya	 Tidak
<u>AWS::Macie::CustomDataIdentifier</u>	 Ya	 Ya	 Tidak
<u>AWS::Macie::FindingsFilter</u>	 Ya	 Ya	 Tidak
<u>AWS::Macie::Session</u>	 Ya	 Ya	 Ya
<u>AWS::MediaConnect::Bridge</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::MediaConnect::BridgeOutput</u>	 Ya	 Ya	 Tidak
<u>AWS::MediaConnect::BridgeSource</u>	 Ya	 Ya	 Tidak
<u>AWS::MediaConnect::Flow</u>	 Ya	 Ya	 Tidak
<u>AWS::MediaConnect::FlowEntitlement</u>	 Ya	 Ya	 Tidak
<u>AWS::MediaConnect::FlowOutput</u>	 Ya	 Ya	 Tidak
<u>AWS::MediaConnect::FlowSource</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::MediaConnect::FlowVpcInterface</u>	 Ya	 Ya	 Ya
<u>AWS::MediaConnect::Gateway</u>	 Ya	 Ya	 Ya
<u>AWS::MediaLive::ChannelPlacementGroup</u>	 Ya	 Ya	 Tidak
<u>AWS::MediaLive::CloudWatchAlarmTemplate</u>	 Ya	 Ya	 Tidak
<u>AWS::MediaLive::CloudWatchAlarmTemplateGroup</u>	 Ya	 Ya	 Tidak
<u>AWS::MediaLive::Cluster</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::MediaLive::EventBridgeRuleTemplate</u>	 Ya	 Ya	 Tidak
<u>AWS::MediaLive::EventBridgeRuleTemplateGroup</u>	 Ya	 Ya	 Tidak
<u>AWS::MediaLive::Multiplex</u>	 Ya	 Ya	 Tidak
<u>AWS::MediaLive::Multiplexprogram</u>	 Ya	 Ya	 Tidak
<u>AWS::MediaLive::Network</u>	 Ya	 Ya	 Tidak
<u>AWS::MediaLive::SdiSource</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::MediaLive::SignalMap</u>	 Ya	 Ya	 Tidak
<u>AWS::MediaPackage::Asset</u>	 Ya	 Ya	 Ya
<u>AWS::MediaPackage::Channel</u>	 Ya	 Ya	 Ya
<u>AWS::MediaPackage::OriginEndpoint</u>	 Ya	 Ya	 Ya
<u>AWS::MediaPackage::PackagingConfiguration</u>	 Ya	 Ya	 Ya
<u>AWS::MediaPackage::PackagingGroup</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::MediaPackageV2::Channel</u>	 Ya	 Ya	 Ya
<u>AWS::MediaPackageV2::ChannelGroup</u>	 Ya	 Ya	 Ya
<u>AWS::MediaPackageV2::ChannelPolicy</u>	 Ya	 Ya	 Tidak
<u>AWS::MediaPackageV2::OriginEndpoint</u>	 Ya	 Ya	 Ya
<u>AWS::MediaPackageV2::OriginEndpointPolicy</u>	 Ya	 Ya	 Tidak
<u>AWS::MediaTailor::Channel</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::MediaTailor::ChannelPolicy</u>	 Ya	 Ya	 Tidak
<u>AWS::MediaTailor::LiveSource</u>	 Ya	 Ya	 Ya
<u>AWS::MediaTailor::PlaybackConfiguration</u>	 Ya	 Ya	 Tidak
<u>AWS::MediaTailor::SourceLocation</u>	 Ya	 Ya	 Ya
<u>AWS::MediaTailor::VodSource</u>	 Ya	 Ya	 Ya
<u>AWS::MemoryDB::ACL</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::MemoryDB::Cluster</u>	 Ya	 Ya	 Ya
<u>AWS::MemoryDB::MultiRegionCluster</u>	 Ya	 Ya	 Tidak
<u>AWS::MemoryDB::ParameterGroup</u>	 Ya	 Ya	 Ya
<u>AWS::MemoryDB::SubnetGroup</u>	 Ya	 Ya	 Ya
<u>AWS::MemoryDB::User</u>	 Ya	 Ya	 Ya
<u>AWS::Neptune::DBCluster</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::Neptune::DBClusterParameterGroup</u>	 Ya	 Ya	 Tidak
<u>AWS::Neptune::DBInstance</u>	 Ya	 Ya	 Tidak
<u>AWS::Neptune::DBParameterGroup</u>	 Ya	 Ya	 Tidak
<u>AWS::Neptune::DBSubnetGroup</u>	 Ya	 Ya	 Tidak
<u>AWS::NeptuneGraph::Graph</u>	 Ya	 Ya	 Tidak
<u>AWS::NeptuneGraph::PrivateGraphEndpoint</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::NetworkFirewall::Firewall</u>	 Ya	 Ya	 Ya
<u>AWS::NetworkFirewall::FirewallPolicy</u>	 Ya	 Ya	 Ya
<u>AWS::NetworkFirewall::LoggingConfiguration</u>	 Ya	 Ya	 Tidak
<u>AWS::NetworkFirewall::RuleGroup</u>	 Ya	 Ya	 Ya
<u>AWS::NetworkFirewall::TLSInspectionConfiguration</u>	 Ya	 Ya	 Tidak
<u>AWS::NetworkFirewall::VpcEndpointAssociation</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::NetworkManager::ConnectAttachment</u>	 Ya	 Ya	 Ya
<u>AWS::NetworkManager::ConnectPeer</u>	 Ya	 Ya	 Ya
<u>AWS::NetworkManager::CoreNetwork</u>	 Ya	 Ya	 Ya
<u>AWS::NetworkManager::CustomerGatewayAssociation</u>	 Ya	 Ya	 Ya
<u>AWS::NetworkManager::Device</u>	 Ya	 Ya	 Ya
<u>AWS::NetworkManager::DirectConnectGatewayAttachment</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::NetworkManager::GlobalNetwork</u>	 Ya	 Ya	 Ya
<u>AWS::NetworkManager::Link</u>	 Ya	 Ya	 Ya
<u>AWS::NetworkManager::LinkAssociation</u>	 Ya	 Ya	 Ya
<u>AWS::NetworkManager::Site</u>	 Ya	 Ya	 Ya
<u>AWS::NetworkManager::SiteToSiteVpnAttachment</u>	 Ya	 Ya	 Ya
<u>AWS::NetworkManager::TransitGatewayPeering</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::NetworkManager::TransitGatewayRegistration</u>	 Ya	 Ya	 Ya
<u>AWS::NetworkManager::TransitGatewayRouteTableAttachment</u>	 Ya	 Ya	 Ya
<u>AWS::NetworkManager::VpcAttachment</u>	 Ya	 Ya	 Ya
<u>AWS::OSIS::Pipeline</u>	 Ya	 Ya	 Ya
<u>AWS::Oam::Link</u>	 Ya	 Ya	 Ya
<u>AWS::Oam::Sink</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::Omics::AnnotationStore</u>	 Ya	 Ya	 Ya
<u>AWS::Omics::ReferenceStore</u>	 Ya	 Ya	 Ya
<u>AWS::Omics::RunGroup</u>	 Ya	 Ya	 Ya
<u>AWS::Omics::SequenceStore</u>	 Ya	 Ya	 Ya
<u>AWS::Omics::VariantStore</u>	 Ya	 Ya	 Ya
<u>AWS::Omics::Workflow</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::Omicss::WorkflowVersion</u>	 Ya	 Ya	 Tidak
<u>AWS::OpenSearchServerless::AccessPolicy</u>	 Ya	 Ya	 Ya
<u>AWS::OpenSearchServerless::Collection</u>	 Ya	 Ya	 Tidak
<u>AWS::OpenSearchServerless::Index</u>	 Ya	 Ya	 Tidak
<u>AWS::OpenSearchServerless::LifecyclePolicy</u>	 Ya	 Ya	 Tidak
<u>AWS::OpenSearchServerless::SecurityConfig</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAC
<u>AWS::OpenSearchServerless::SecurityPolicy</u>	 Ya	 Ya	 Ya
<u>AWS::OpenSearchServerless::VpcEndpoint</u>	 Ya	 Ya	 Ya
<u>AWS::OpenSearchService::Application</u>	 Ya	 Ya	 Tidak
<u>AWS::OpenSearchService::Domain</u>	 Ya	 Ya	 Tidak
<u>AWS::Organizations::Account</u>	 Ya	 Ya	 Ya
<u>AWS::Organizations::Organization</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::Organizations::OrganizationalUnit</u>	 Ya	 Ya	 Ya
<u>AWS::Organizations::Policy</u>	 Ya	 Ya	 Ya
<u>AWS::Organizations::ResourcePolicy</u>	 Ya	 Ya	 Ya
<u>AWS::PCAConectorAD::Connector</u>	 Ya	 Ya	 Ya
<u>AWS::PCAConectorAD::DirectoryRegistration</u>	 Ya	 Ya	 Ya
<u>AWS::PCAConectorAD::ServicePrincipalName</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAC
<u>AWS::PCAConnectorAD::Template</u>	 Ya	 Ya	 Ya
<u>AWS::PCAConnectorAD::TemplateGroupAccessControlEntry</u>	 Ya	 Ya	 Ya
<u>AWS::PCAConnectorSCEP::Challenge</u>	 Ya	 Ya	 Tidak
<u>AWS::PCAConnectorSCEP::Connector</u>	 Ya	 Ya	 Tidak
<u>AWS::PCS::Cluster</u>	 Ya	 Ya	 Tidak
<u>AWS::PCS::ComputeNodeGroup</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAC
<u>AWS::PCS::Queue</u>	 Ya	 Ya	 Tidak
<u>AWS::Panorama::ApplicationInstance</u>	 Ya	 Ya	 Ya
<u>AWS::Panorama::Package</u>	 Ya	 Ya	 Ya
<u>AWS::Panorama::PackageVersion</u>	 Ya	 Ya	 Tidak
<u>AWS::PaymentCryptography::Alias</u>	 Ya	 Ya	 Tidak
<u>AWS::PaymentCryptography::Key</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::Personalize::Dataset</u>	 Ya	 Ya	 Ya
<u>AWS::Personalize::DatasetGroup</u>	 Ya	 Ya	 Ya
<u>AWS::Personalize::Schema</u>	 Ya	 Ya	 Ya
<u>AWS::Personalize::Solution</u>	 Ya	 Ya	 Tidak
<u>AWS::Pinpoint::InAppTemplate</u>	 Ya	 Ya	 Ya
<u>AWS::Pipes::Pipe</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAC
<u>AWS::Proton::EnvironmentAccountConnection</u>	 Ya	 Ya	 Ya
<u>AWS::Proton::EnvironmentTemplate</u>	 Ya	 Ya	 Ya
<u>AWS::Proton::ServiceTemplate</u>	 Ya	 Ya	 Ya
<u>AWS::QBusiness::Application</u>	 Ya	 Ya	 Tidak
<u>AWS::QBusiness::DataAccessor</u>	 Ya	 Ya	 Tidak
<u>AWS::QBusiness::DataSource</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::QBusiness::Index</u>	 Ya	 Ya	 Tidak
<u>AWS::QBusiness::Permission</u>	 Ya	 Ya	 Tidak
<u>AWS::QBusiness::Plugin</u>	 Ya	 Ya	 Tidak
<u>AWS::QBusiness::Retriever</u>	 Ya	 Ya	 Tidak
<u>AWS::QBusiness::WebExperience</u>	 Ya	 Ya	 Tidak
<u>AWS::QLDB::Stream</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::QuickSight::Analysis</u>	 Ya	 Ya	 Tidak
<u>AWS::QuickSight::CustomPermissions</u>	 Ya	 Ya	 Tidak
<u>AWS::QuickSight::Dashboard</u>	 Ya	 Ya	 Tidak
<u>AWS::QuickSight::DataSet</u>	 Ya	 Ya	 Ya
<u>AWS::QuickSight::DataSource</u>	 Ya	 Ya	 Ya
<u>AWS::QuickSight::Folder</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::QuickSight::RefreshSchedule</u>	 Ya	 Ya	 Ya
<u>AWS::QuickSight::Template</u>	 Ya	 Ya	 Tidak
<u>AWS::QuickSight::Theme</u>	 Ya	 Ya	 Tidak
<u>AWS::QuickSight::Topic</u>	 Ya	 Ya	 Ya
<u>AWS::QuickSight::VPCConnection</u>	 Ya	 Ya	 Ya
<u>AWS::RAM::Permission</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::RAM::ResourceShare</u>	 Ya	 Ya	 Tidak
<u>AWS::RDS::CustomDBEngineVersion</u>	 Ya	 Ya	 Tidak
<u>AWS::RDS::DBCluster</u>	 Ya	 Ya	 Ya
<u>AWS::RDS::DBClusterParameterGroup</u>	 Ya	 Ya	 Tidak
<u>AWS::RDS::DBInstance</u>	 Ya	 Tidak	 Ya
<u>AWS::RDS::DBParameterGroup</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::RDS::DBProxy</u>	 Ya	 Ya	 Ya
<u>AWS::RDS::DBProxyEndpoint</u>	 Ya	 Ya	 Ya
<u>AWS::RDS::DBProxyTargetGroup</u>	 Ya	 Ya	 Ya
<u>AWS::RDS::DBShardGroup</u>	 Ya	 Ya	 Tidak
<u>AWS::RDS::DBSubnetGroup</u>	 Ya	 Ya	 Ya
<u>AWS::RDS::EventSubscription</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::RDS::GlobalCluster</u>	 Ya	 Ya	 Ya
<u>AWS::RDS::Integration</u>	 Ya	 Ya	 Tidak
<u>AWS::RDS::OptionGroup</u>	 Ya	 Ya	 Ya
<u>AWS::RUM::AppMonitor</u>	 Ya	 Ya	 Ya
<u>AWS::Rbin::Rule</u>	 Ya	 Ya	 Tidak
<u>AWS::Redshift::Cluster</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::Redshift::ClusterParameterGroup</u>	 Ya	 Ya	 Tidak
<u>AWS::Redshift::ClusterSubnetGroup</u>	 Ya	 Ya	 Tidak
<u>AWS::Redshift::EndpointAccess</u>	 Ya	 Ya	 Ya
<u>AWS::Redshift::EndpointAuthorization</u>	 Ya	 Ya	 Ya
<u>AWS::Redshift::EventSubscription</u>	 Ya	 Ya	 Tidak
<u>AWS::Redshift::Integration</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAC
<u>AWS::Redshift::ScheduledAction</u>	 Ya	 Ya	 Ya
<u>AWS::RedshiftServerless::Namespace</u>	 Ya	 Ya	 Tidak
<u>AWS::RedshiftServerless::Snapshot</u>	 Ya	 Ya	 Tidak
<u>AWS::RedshiftServerless::Workgroup</u>	 Ya	 Ya	 Tidak
<u>AWS::RefactorSpaces::Application</u>	 Ya	 Ya	 Ya
<u>AWS::RefactorSpaces::Environment</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::RefactorSpaces::Route</u>	 Ya	 Ya	 Ya
<u>AWS::RefactorSpaces::Service</u>	 Ya	 Ya	 Ya
<u>AWS::Rekognition::Collection</u>	 Ya	 Ya	 Ya
<u>AWS::Rekognition::Project</u>	 Ya	 Ya	 Ya
<u>AWS::Rekognition::StreamProcessor</u>	 Ya	 Ya	 Ya
<u>AWS::ResilienceHub::App</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::ResilienceHub::ResiliencyPolicy</u>	 Ya	 Ya	 Ya
<u>AWS::ResourceExplorer2::DefaultViewAssociation</u>	 Ya	 Ya	 Tidak
<u>AWS::ResourceExplorer2::Index</u>	 Ya	 Ya	 Ya
<u>AWS::ResourceExplorer2::View</u>	 Ya	 Ya	 Ya
<u>AWS::ResourceGroups::Group</u>	 Ya	 Ya	 Ya
<u>AWS::ResourceGroups::TagSyncTask</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::RoboMaker::Fleet</u>	 Ya	 Ya	 Tidak
<u>AWS::RoboMaker::Robot</u>	 Ya	 Ya	 Tidak
<u>AWS::RoboMaker::RobotApplication</u>	 Ya	 Ya	 Tidak
<u>AWS::RoboMaker::RobotApplicationVersion</u>	 Ya	 Ya	 Tidak
<u>AWS::RoboMaker::SimulationApplication</u>	 Ya	 Ya	 Tidak
<u>AWS::RoboMaker::SimulationApplicationVersion</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::RolesAnywhere::CRL</u>	 Ya	 Ya	 Ya
<u>AWS::RolesAnywhere::Profile</u>	 Ya	 Ya	 Ya
<u>AWS::RolesAnywhere::TrustAnchor</u>	 Ya	 Ya	 Tidak
<u>AWS::Route53::CidrCollection</u>	 Ya	 Ya	 Ya
<u>AWS::Route53::DNSSEC</u>	 Ya	 Ya	 Ya
<u>AWS::Route53::HealthCheck</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::Route53::HostedZone</u>	 Ya	 Ya	 Ya
<u>AWS::Route53::KeySigningKey</u>	 Ya	 Ya	 Ya
<u>AWS::Route53Profiles::Profile</u>	 Ya	 Ya	 Tidak
<u>AWS::Route53Profiles::ProfileAssociation</u>	 Ya	 Ya	 Tidak
<u>AWS::Route53Profiles::ProfileResourceAssociation</u>	 Ya	 Ya	 Tidak
<u>AWS::Route53RecoveryControl::Cluster</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::Route53RecoveryControl::ControlPanel</u>	 Ya	 Ya	 Ya
<u>AWS::Route53RecoveryControl::RoutingControl</u>	 Ya	 Ya	 Ya
<u>AWS::Route53RecoveryControl::SafetyRule</u>	 Ya	 Ya	 Ya
<u>AWS::Route53RecoveryReadiness::Cell</u>	 Ya	 Ya	 Ya
<u>AWS::Route53RecoveryReadiness::ReadinessCheck</u>	 Ya	 Ya	 Ya
<u>AWS::Route53RecoveryReadiness::RecoveryGroup</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAC
<u>AWS::Route53RecoveryReadiness::ResourceSet</u>	 Ya	 Ya	 Ya
<u>AWS::Route53Resolver::FirewallDomainList</u>	 Ya	 Ya	 Tidak
<u>AWS::Route53Resolver::FirewallRuleGroup</u>	 Ya	 Ya	 Ya
<u>AWS::Route53Resolver::FirewallRuleGroupAssociation</u>	 Ya	 Ya	 Ya
<u>AWS::Route53Resolver::OutpostResolver</u>	 Ya	 Ya	 Ya
<u>AWS::Route53Resolver::ResolverConfig</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAC
<u>AWS::Route53Resolver::ResolverDNSSECConfig</u>	 Ya	 Ya	 Ya
<u>AWS::Route53Resolver::ResolverEndpoint</u>	 Ya	 Ya	 Tidak
<u>AWS::Route53Resolver::ResolverQueryLoggingConfig</u>	 Ya	 Ya	 Ya
<u>AWS::Route53Resolver::ResolverQueryLoggingConfigAssociation</u>	 Ya	 Ya	 Ya
<u>AWS::Route53Resolver::ResolverRule</u>	 Ya	 Ya	 Tidak
<u>AWS::Route53Resolver::ResolverRuleAssociation</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::S3::AccessGrant</u>	 Ya	 Ya	 Tidak
<u>AWS::S3::AccessGrantsInstance</u>	 Ya	 Ya	 Tidak
<u>AWS::S3::AccessGrantsLocation</u>	 Ya	 Ya	 Tidak
<u>AWS::S3::AccessPoint</u>	 Ya	 Ya	 Ya
<u>AWS::S3::Bucket</u>	 Ya	 Ya	 Ya
<u>AWS::S3::BucketPolicy</u>	 Ya	 Tidak	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAC
<u>AWS::S3::MultiRegionAccessPoint</u>	 Ya	 Ya	 Ya
<u>AWS::S3::MultiRegionAccessPointPolicy</u>	 Ya	 Tidak	 Tidak
<u>AWS::S3::StorageLens</u>	 Ya	 Ya	 Ya
<u>AWS::S3::StorageLensGroup</u>	 Ya	 Ya	 Tidak
<u>AWS::S3Express::AccessPoint</u>	 Ya	 Ya	 Tidak
<u>AWS::S3Express::BucketPolicy</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::S3Express::DirectoryBucket</u>	 Ya	 Ya	 Tidak
<u>AWS::S3ObjectLambda::AccessPoint</u>	 Ya	 Ya	 Ya
<u>AWS::S3ObjectLambda::AccessPointPolicy</u>	 Ya	 Ya	 Tidak
<u>AWS::S3Outposts::AccessPoint</u>	 Ya	 Ya	 Ya
<u>AWS::S3Outposts::Bucket</u>	 Ya	 Ya	 Ya
<u>AWS::S3Outposts::BucketPolicy</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::S3Outposts::Endpoint</u>	 Ya	 Ya	 Ya
<u>AWS::S3Tables::Namespace</u>	 Ya	 Ya	 Tidak
<u>AWS::S3Tables::TableBucket</u>	 Ya	 Ya	 Tidak
<u>AWS::S3Tables::TableBucketPolicy</u>	 Ya	 Ya	 Tidak
<u>AWS::SES::ConfigurationSet</u>	 Ya	 Ya	 Tidak
<u>AWS::SES::ConfigurationSetEventDestination</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::SES::ContactList</u>	 Ya	 Ya	 Tidak
<u>AWS::SES::DedicatedIpPool</u>	 Ya	 Ya	 Ya
<u>AWS::SES::EmailIdentity</u>	 Ya	 Ya	 Ya
<u>AWS::SES::MailManagerAddonInstance</u>	 Ya	 Ya	 Tidak
<u>AWS::SES::MailManagerAddonSubscription</u>	 Ya	 Ya	 Tidak
<u>AWS::SES::MailManagerAddressList</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::SES::MailManagerArchive</u>	 Ya	 Ya	 Tidak
<u>AWS::SES::MailManagerIngressPoint</u>	 Ya	 Ya	 Tidak
<u>AWS::SES::MailManagerRelay</u>	 Ya	 Ya	 Tidak
<u>AWS::SES::MailManagerRuleSet</u>	 Ya	 Ya	 Tidak
<u>AWS::SES::MailManagerTrafficPolicy</u>	 Ya	 Ya	 Tidak
<u>AWS::SES::Template</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::SES::VdmAttributes</u>	 Ya	 Ya	 Tidak
<u>AWS::SNS::Subscription</u>	 Ya	 Ya	 Tidak
<u>AWS::SNS::Topic</u>	 Ya	 Ya	 Ya
<u>AWS::SNS::TopicInlinePolicy</u>	 Ya	 Ya	 Tidak
<u>AWS::SQS::Queue</u>	 Ya	 Ya	 Ya
<u>AWS::SQS::QueueInlinePolicy</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::SSM::Association</u>	 Ya	 Ya	 Ya
<u>AWS::SSM::Document</u>	 Ya	 Tidak	 Tidak
<u>AWS::SSM::Parameter</u>	 Ya	 Ya	 Tidak
<u>AWS::SSM::PatchBaseline</u>	 Ya	 Ya	 Tidak
<u>AWS::SSM::ResourceDataSync</u>	 Ya	 Ya	 Tidak
<u>AWS::SSM::ResourcePolicy</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::SSMContacts::Contact</u>	 Ya	 Ya	 Ya
<u>AWS::SSMContacts::ContactChannel</u>	 Ya	 Ya	 Ya
<u>AWS::SSMContacts::Plan</u>	 Ya	 Ya	 Tidak
<u>AWS::SSMContacts::Rotation</u>	 Ya	 Ya	 Tidak
<u>AWS::SSMGuiConnect::Preferences</u>	 Ya	 Ya	 Tidak
<u>AWS::SSMIncidents::ReplicationSet</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::SSMIncidents::ResponsePlan</u>	 Ya	 Ya	 Ya
<u>AWS::SSMQuickSetup::ConfigurationManager</u>	 Ya	 Ya	 Tidak
<u>AWS::SSO::Application</u>	 Ya	 Ya	 Tidak
<u>AWS::SSO::ApplicationAssignment</u>	 Ya	 Ya	 Tidak
<u>AWS::SSO::Assignment</u>	 Ya	 Ya	 Tidak
<u>AWS::SSO::Instance</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAC
<u>AWS::SSO::InstanceAccessControlAttributeConfiguration</u>	 Ya	 Ya	 Ya
<u>AWS::SageMaker::App</u>	 Ya	 Ya	 Tidak
<u>AWS::SageMaker::AppImageConfig</u>	 Ya	 Ya	 Tidak
<u>AWS::SageMaker::Cluster</u>	 Ya	 Ya	 Tidak
<u>AWS::SageMaker::DataQualityJobDefinition</u>	 Ya	 Ya	 Tidak
<u>AWS::SageMaker::Device</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::SageMaker::DeviceFleet</u>	 Ya	 Ya	 Tidak
<u>AWS::SageMaker::Domain</u>	 Ya	 Ya	 Tidak
<u>AWS::SageMaker::Endpoint</u>	 Ya	 Ya	 Tidak
<u>AWS::SageMaker::FeatureGroup</u>	 Ya	 Ya	 Ya
<u>AWS::SageMaker::Image</u>	 Ya	 Ya	 Ya
<u>AWS::SageMaker::ImageVersion</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::SageMaker::InferenceComponent</u>	 Ya	 Ya	 Tidak
<u>AWS::SageMaker::InferenceExperiment</u>	 Ya	 Ya	 Ya
<u>AWS::SageMaker::MlflowTrackingServer</u>	 Ya	 Ya	 Tidak
<u>AWS::SageMaker::ModelBiasJobDefinition</u>	 Ya	 Ya	 Tidak
<u>AWS::SageMaker::ModelCard</u>	 Ya	 Ya	 Ya
<u>AWS::SageMaker::ModelExplainabilityJobDefinition</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::SageMaker::ModelPackage</u>	 Ya	 Ya	 Ya
<u>AWS::SageMaker::ModelPackageGroup</u>	 Ya	 Ya	 Ya
<u>AWS::SageMaker::ModelQualityJobDefinition</u>	 Ya	 Ya	 Tidak
<u>AWS::SageMaker::MonitoringSchedule</u>	 Ya	 Ya	 Tidak
<u>AWS::SageMaker::PartnerApp</u>	 Ya	 Ya	 Tidak
<u>AWS::SageMaker::Pipeline</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAC
<u>AWS::SageMaker::Project</u>	 Ya	 Ya	 Ya
<u>AWS::SageMaker::Space</u>	 Ya	 Ya	 Tidak
<u>AWS::SageMaker::StudioLifecycleConfig</u>	 Ya	 Ya	 Tidak
<u>AWS::SageMaker::UserProfile</u>	 Ya	 Ya	 Tidak
<u>AWS::Scheduler::Schedule</u>	 Ya	 Ya	 Ya
<u>AWS::Scheduler::ScheduleGroup</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::SecretsManager::ResourcePolicy</u>	 Ya	 Tidak	 Tidak
<u>AWS::SecretsManager::RotationSchedule</u>	 Ya	 Ya	 Tidak
<u>AWS::SecretsManager::Secret</u>	 Ya	 Ya	 Ya
<u>AWS::SecretsManager::SecretTargetAttachment</u>	 Ya	 Ya	 Tidak
<u>AWS::SecurityHub::AggregatorV2</u>	 Ya	 Ya	 Tidak
<u>AWS::SecurityHub::AutomationRule</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAC
<u>AWS::SecurityHub::AutomationRuleV2</u>	 Ya	 Ya	 Tidak
<u>AWS::SecurityHub::ConfigurationPolicy</u>	 Ya	 Ya	 Tidak
<u>AWS::SecurityHub::DelegatedAdmin</u>	 Ya	 Ya	 Tidak
<u>AWS::SecurityHub::FindingAggregator</u>	 Ya	 Ya	 Tidak
<u>AWS::SecurityHub::Hub</u>	 Ya	 Ya	 Tidak
<u>AWS::SecurityHub::HubV2</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::SecurityHub::Insight</u>	 Ya	 Ya	 Tidak
<u>AWS::SecurityHub::OrganizationConfiguration</u>	 Ya	 Ya	 Tidak
<u>AWS::SecurityHub::PolicyAssociation</u>	 Ya	 Ya	 Tidak
<u>AWS::SecurityHub::ProductSubscription</u>	 Ya	 Ya	 Tidak
<u>AWS::SecurityHub::SecurityControl</u>	 Ya	 Ya	 Tidak
<u>AWS::SecurityHub::Standard</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::SecurityLake::AwsLogSource</u>	 Ya	 Ya	 Tidak
<u>AWS::SecurityLake::DataLake</u>	 Ya	 Ya	 Tidak
<u>AWS::SecurityLake::Subscriber</u>	 Ya	 Ya	 Tidak
<u>AWS::SecurityLake::SubscriberNotification</u>	 Ya	 Ya	 Tidak
<u>AWS::ServiceCatalog::CloudFormationProvisionedProduct</u>	 Ya	 Tidak	 Tidak
<u>AWS::ServiceCatalog::ServiceAction</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::ServiceCatalog::ServiceActionAssociation</u>	 Ya	 Ya	 Tidak
<u>AWS::ServiceCatalogAppRegistry::Application</u>	 Ya	 Ya	 Ya
<u>AWS::ServiceCatalogAppRegistry::AttributeGroup</u>	 Ya	 Ya	 Ya
<u>AWS::ServiceCatalogAppRegistry::AttributeGroupAssociation</u>	 Ya	 Ya	 Tidak
<u>AWS::ServiceCatalogAppRegistry::ResourceAssociation</u>	 Ya	 Tidak	 Tidak
<u>AWS::Shield::DRTAccess</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::Shield::ProactiveEngagement</u>	 Ya	 Ya	 Tidak
<u>AWS::Shield::Protection</u>	 Ya	 Ya	 Tidak
<u>AWS::Shield::ProtectionGroup</u>	 Ya	 Ya	 Tidak
<u>AWS::Signer::ProfilePermission</u>	 Ya	 Ya	 Tidak
<u>AWS::Signer::SigningProfile</u>	 Ya	 Ya	 Ya
<u>AWS::SimSpaceWeaver::Simulation</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::StepFunctions::Activity</u>	 Ya	 Ya	 Tidak
<u>AWS::StepFunctions::StateMachine</u>	 Ya	 Ya	 Ya
<u>AWS::StepFunctions::StateMachineAlias</u>	 Ya	 Ya	 Ya
<u>AWS::StepFunctions::StateMachineVersion</u>	 Ya	 Ya	 Ya
<u>AWS::SupportApp::AccountAlias</u>	 Ya	 Ya	 Ya
<u>AWS::SupportApp::SlackChannelConfiguration</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::SupportApp::SlackWorkspaceConfiguration</u>	 Ya	 Ya	 Ya
<u>AWS::Synthetics::Canary</u>	 Ya	 Ya	 Ya
<u>AWS::Synthetics::Group</u>	 Ya	 Ya	 Ya
<u>AWS::SystemsManagerSAP::Application</u>	 Ya	 Ya	 Tidak
<u>AWS::Timestream::Database</u>	 Ya	 Ya	 Ya
<u>AWS::Timestream::InfluxDBInstance</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::Timestream::ScheduledQuery</u>	 Ya	 Ya	 Tidak
<u>AWS::Timestream::Table</u>	 Ya	 Ya	 Ya
<u>AWS::Transfer::Agreement</u>	 Ya	 Ya	 Ya
<u>AWS::Transfer::Certificate</u>	 Ya	 Ya	 Ya
<u>AWS::Transfer::Connector</u>	 Ya	 Ya	 Ya
<u>AWS::Transfer::Profile</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::Transfer::Server</u>	 Ya	 Ya	 Tidak
<u>AWS::Transfer::User</u>	 Ya	 Ya	 Tidak
<u>AWS::Transfer::WebApp</u>	 Ya	 Ya	 Tidak
<u>AWS::Transfer::Workflow</u>	 Ya	 Ya	 Ya
<u>AWS::VerifiedPermissions::IdentitySource</u>	 Ya	 Ya	 Ya
<u>AWS::VerifiedPermissions::Policy</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::VerifiedPermissions::PolicyStore</u>	 Ya	 Ya	 Ya
<u>AWS::VerifiedPermissions::PolicyTemplate</u>	 Ya	 Ya	 Ya
<u>AWS::VoiceID::Domain</u>	 Ya	 Ya	 Tidak
<u>AWS::VpcLattice::AccessLogSubscription</u>	 Ya	 Ya	 Ya
<u>AWS::VpcLattice::AuthPolicy</u>	 Ya	 Ya	 Tidak
<u>AWS::VpcLattice::Listener</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::VpcLattice::ResourceConfiguration</u>	 Ya	 Ya	 Tidak
<u>AWS::VpcLattice::ResourceGateway</u>	 Ya	 Ya	 Tidak
<u>AWS::VpcLattice::ResourcePolicy</u>	 Ya	 Ya	 Tidak
<u>AWS::VpcLattice::Rule</u>	 Ya	 Ya	 Ya
<u>AWS::VpcLattice::Service</u>	 Ya	 Ya	 Ya
<u>AWS::VpcLattice::ServiceNetwork</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::VpcLattice::ServiceNetworkResourceAssociation</u>	 Ya	 Ya	 Tidak
<u>AWS::VpcLattice::ServiceNetworkServiceAssociation</u>	 Ya	 Ya	 Ya
<u>AWS::VpcLattice::ServiceNetworkVpcAssociation</u>	 Ya	 Ya	 Ya
<u>AWS::VpcLattice::TargetGroup</u>	 Ya	 Ya	 Ya
<u>AWS::WAFv2::IPSet</u>	 Ya	 Ya	 Ya
<u>AWS::WAFv2::LoggingConfiguration</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::WAFv2::RegexPatternSet</u>	 Ya	 Ya	 Ya
<u>AWS::WAFv2::RuleGroup</u>	 Ya	 Ya	 Ya
<u>AWS::WAFv2::WebACL</u>	 Ya	 Ya	 Ya
<u>AWS::WAFv2::WebACLAssociation</u>	 Ya	 Ya	 Tidak
<u>AWS::Wisdom::AIAgent</u>	 Ya	 Ya	 Tidak
<u>AWS::Wisdom::AIAgentVersion</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::Wisdom::AIGuardrail</u>	 Ya	 Ya	 Tidak
<u>AWS::Wisdom::AIGuardrailVersion</u>	 Ya	 Ya	 Tidak
<u>AWS::Wisdom::AIPrompt</u>	 Ya	 Ya	 Tidak
<u>AWS::Wisdom::AIPromptVersion</u>	 Ya	 Ya	 Tidak
<u>AWS::Wisdom::Assistant</u>	 Ya	 Ya	 Ya
<u>AWS::Wisdom::AssistantAssociation</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::Wisdom::KnowledgeBase</u>	 Ya	 Ya	 Ya
<u>AWS::Wisdom::MessageTemplate</u>	 Ya	 Ya	 Tidak
<u>AWS::Wisdom::MessageTemplateVersion</u>	 Ya	 Ya	 Tidak
<u>AWS::WorkSpaces::ConnectionAlias</u>	 Ya	 Ya	 Tidak
<u>AWS::WorkSpaces::WorkspacesPool</u>	 Ya	 Ya	 Tidak
<u>AWS::WorkSpacesThinClient::Environment</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::WorkSpacesWeb::BrowserSettings</u>	 Ya	 Ya	 Ya
<u>AWS::WorkSpacesWeb::DataProtectionSettings</u>	 Ya	 Ya	 Tidak
<u>AWS::WorkSpacesWeb::IdentityProvider</u>	 Ya	 Ya	 Ya
<u>AWS::WorkSpacesWeb::IpAccessSettings</u>	 Ya	 Ya	 Ya
<u>AWS::WorkSpacesWeb::NetworkSettings</u>	 Ya	 Ya	 Ya
<u>AWS::WorkSpacesWeb::Portal</u>	 Ya	 Ya	 Ya

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
<u>AWS::WorkSpacesWeb::TrustStore</u>	 Ya	 Ya	 Ya
<u>AWS::WorkSpacesWeb::UserAccessLoggingSettings</u>	 Ya	 Ya	 Ya
<u>AWS::WorkSpacesWeb::UserSettings</u>	 Ya	 Ya	 Ya
<u>AWS::XRay::Group</u>	 Ya	 Ya	 Ya
<u>AWS::XRay::ResourcePolicy</u>	 Ya	 Ya	 Ya
<u>AWS::XRay::SamplingRule</u>	 Ya	 Ya	 Tidak

Sumber Daya	Impor	Deteksi penyimpanan	Generator IAc
AWS::XRay::TransactionSearchConfig	 Ya	 Ya	 Tidak

Gunakan tautan buat cepat untuk membuat tumpukan CloudFormation

Tautan buat cepat menyediakan metode yang disederhanakan untuk meluncurkan CloudFormation tumpukan langsung dari URLs konsol. CloudFormation Dengan menentukan URL template, nama tumpukan, dan parameter template sebagai parameter kueri URL, Anda dapat mengisi satu halaman Buat tumpukan dan mempercepat pembuatan tumpukan. Ini menyederhanakan proses pembuatan tumpukan dengan mengurangi jumlah halaman wizard dan jumlah input pengguna yang diperlukan. Ini juga mengoptimalkan penggunaan kembali template, karena Anda dapat membuat beberapa URLs yang menentukan nilai berbeda untuk template yang sama.

Format URL

Tautan cepat-buat mengikuti format URL ini:

```
https://region-code.console.aws.amazon.com/cloudformation/home?region=region-code#/stacks/create/review
?templateURL=TemplateURL
&stackName=StackName
&param_parameterName=parameterValue
```

CloudFormation mendukung parameter kueri URL berikut:

URL Templat

Wajib. `templateURLParameter` menentukan URL template tumpukan yang terletak di bucket Amazon S3. Untuk menghindari masalah akses dengan URL S3 yang telah ditetapkan sebelumnya, pastikan Anda menyandikan URL URL.

Format URL S3 yang didukung:

- `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`(format warisan)

Nama tumpukan

Opsional. Gunakan `stackName` parameter untuk menentukan nama CloudFormation tumpukan yang akan dibuat. Nama tumpukan hanya dapat berisi karakter alfanumerik (peka huruf besar/kecil) dan tanda hubung. Ini harus dimulai dengan karakter abjad dan tidak boleh lebih dari 128 karakter.

Parameter template

Opsional. Untuk parameter dalam template tumpukan yang bukan tipe `NoEcho` parameter, gunakan format `param_parameterName` dalam string kueri URL. Parameter URL harus menyertakan prefiks `param_`, dan segmen nama parameter harus sama persis dengan nama parameter di templat. Sebagai contoh: `param_DBName`.

CloudFormation mengabaikan parameter yang tidak ada di template, dan parameter apa pun yang ditentukan dengan `NoEcho` propertinya disetel ke `true` tipe (biasanya, nama pengguna dan kata sandi). Parameter URL menerima nilai default yang ditentukan di templat. Anda dapat memasukkan sebanyak mungkin parameter yang diperlukan.

Important

Daripada menyematkan informasi sensitif secara langsung di CloudFormation template Anda, kami sarankan Anda menggunakan parameter dinamis dalam template tumpukan untuk mereferensikan informasi sensitif yang disimpan dan dikelola di luar CloudFormation, seperti di AWS Systems Manager Parameter Store atau AWS Secrets Manager.

Untuk informasi selengkapnya, lihat [Do not embed credentials dalam praktik terbaik template Anda](#).

Semua nama parameter kueri memperhatikan huruf besar atau kecil. Pengguna dapat menerima nilai ini di konsol sebelum membuat tumpukan.

Contoh

Contoh berikut didasarkan pada contoh [contoh template WordPress dasar tunggal](#). String kueri mencakup parameter `templateURL` yang diperlukan dan parameter `stackName`, `DBName`, `InstanceType`, dan `KeyName`.

URL berikut memiliki jeda baris yang ditambahkan untuk kejelasan.

```
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
```

URL berikut mencakup parameter yang sama seperti contoh sebelumnya, tetapi jeda baris dihapus. Ini adalah format URL aktual.

```
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=
```

Membuat tumpukan menggunakan tautan buat cepat

Saat Anda membuka tautan buat cepat, Anda diarahkan ke konsol. CloudFormation Konsol terbuka langsung ke halaman tumpukan Quick create, dengan nilai yang diberikan secara otomatis digunakan untuk parameter.

Untuk membuat tumpukan menggunakan tautan buat cepat (konsol)

1. Pada halaman Quick create stack, untuk Template, Template URL, konfirmasi URL template sudah benar.
2. Perluas bagian Lihat template untuk memverifikasi template.
3. Untuk nama Stack, verifikasi nama tumpukan yang telah diisi sebelumnya.
4. Tinjau bagian Parameter. Verifikasi bahwa nilai parameter yang sudah terisi sudah benar. Isi parameter yang diperlukan yang tidak ditentukan dalam URL. Ubah nilai yang sudah terisi sebelumnya jika diperlukan.

5. Selanjutnya, Anda dapat mengkonfigurasi pengaturan berikut:

- Tag - Mengatur sumber daya dengan pasangan kunci-nilai.
- Izin - Pilih peran layanan IAM untuk operasi tumpukan.
- Opsi kegagalan tumpukan - Pilih untuk memutar kembali (default) atau mempertahankan sumber daya.
- Kebijakan tumpukan - Kontrol izin pembaruan sumber daya.
- Konfigurasi rollback - Konfigurasikan rollback berbasis CloudWatch alarm.
- Opsi pemberitahuan - Siapkan notifikasi Amazon SNS untuk acara tumpukan.
- Opsi pembuatan tumpukan — Tentukan waktu pembuatan tumpukan maksimum dan aktifkan perlindungan penghentian untuk mencegah penghapusan yang tidak disengaja.

Untuk informasi selengkapnya, lihat [Konfigurasikan opsi tumpukan](#).

6. Untuk Kemampuan, lengkapi semua pengakuan yang diperlukan. Misalnya, jika template Anda berisi sumber daya IAM, pilih Saya mengakui bahwa template ini dapat membuat sumber daya IAM untuk menentukan bahwa Anda ingin menggunakan sumber daya IAM dalam template. Untuk informasi selengkapnya, lihat [Mengakui sumber daya IAM dalam templat CloudFormation](#).
7. (Opsional) Anda dapat membuat set perubahan untuk melihat pratinjau konfigurasi tumpukan sebelum membuatnya. Pilih Buat set perubahan dan ikuti petunjuknya. Untuk informasi selengkapnya, lihat [Pratinjau konfigurasi tumpukan Anda](#).
8. Saat Anda siap, pilih Buat tumpukan untuk meluncurkan tumpukan dan kemudian pantau kemajuan pembuatan tumpukan di tab Acara. Lihat informasi yang lebih lengkap di [Pantau kemajuan tumpukan](#).

Contoh perintah operasi CloudFormation tumpukan untuk AWS CLI dan PowerShell

Contoh baris perintah berikut menunjukkan bagaimana melakukan CloudFormation tindakan individu dengan AWS CLI dan PowerShell. Contoh-contoh ini hanya mencakup tindakan yang paling umum digunakan. Untuk daftar lengkap, lihat [cloudformation](#) di AWS CLI Command Reference.

Contoh dalam panduan ini menggunakan konvensi garis miring terbalik (\) untuk menunjukkan bahwa baris perintah panjang berlanjut pada baris berikutnya.

Topik

- [Batalkan pembaruan tumpukan](#)
- [Terus mengembalikan pembaruan](#)
- [Buat tumpukan](#)
- [Buat tumpukan yang mencakup transformasi](#)
- [Hapus tumpukan](#)
- [Jelaskan peristiwa tumpukan](#)
- [Jelaskan sumber daya tumpukan](#)
- [Jelaskan sumber daya tumpukan](#)
- [Jelaskan tumpukan](#)
- [Dapatkan template](#)
- [Daftar sumber daya tumpukan](#)
- [Daftar tumpukan](#)
- [Perbarui tumpukan](#)
- [Validasi template Anda](#)
- [Unggah artefak lokal ke bucket S3 dengan AWS CLI](#)

Batalkan pembaruan tumpukan

Gunakan perintah [cancel-update-stack](#) perintah untuk membatalkan pembaruan tumpukan. Untuk informasi selengkapnya, lihat [Batalkan pembaruan tumpukan](#).

CLI

AWS CLI

Untuk membatalkan pembaruan tumpukan yang sedang berlangsung

`cancel-update-stack` Perintah berikut membatalkan pembaruan tumpukan pada `myteststack` tumpukan:

```
aws cloudformation cancel-update-stack --stack-name myteststack
```

- Untuk detail API, lihat [CancelUpdateStack](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell V4

Contoh 1: Membatalkan pembaruan pada tumpukan yang ditentukan.

```
Stop-CFNUpdateStack -StackName "myStack"
```

- Untuk detail API, lihat [CancelUpdateStack](#) di Referensi Alat AWS untuk PowerShell Cmdlet (V4).

Alat untuk PowerShell V5

Contoh 1: Membatalkan pembaruan pada tumpukan yang ditentukan.

```
Stop-CFNUpdateStack -StackName "myStack"
```

- Untuk detail API, lihat [CancelUpdateStack](#) di Referensi Alat AWS untuk PowerShell Cmdlet (V5).

Terus mengembalikan pembaruan

Gunakan [continue-update-rollback](#) perintah untuk melanjutkan memutar kembali pembaruan. Untuk informasi selengkapnya, lihat [Terus mengembalikan pembaruan](#).

CLI

AWS CLI

Untuk mencoba kembali rollback pembaruan

`continue-update-rollback` Contoh berikut melanjutkan operasi rollback dari pembaruan tumpukan yang sebelumnya gagal.

```
aws cloudformation continue-update-rollback \  
  --stack-name my-stack
```

Perintah ini tidak menghasilkan output.

- Untuk detail API, lihat [ContinueUpdateRollback](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell V4

Contoh 1: Melanjutkan rollback dari tumpukan bernama, yang seharusnya dalam status 'UPDATE_ROLLBACK_FAILED'. Jika rollback lanjutan berhasil, tumpukan akan memasukkan status 'UPDATE_ROLLBACK_COMPLETE'.

```
Resume-CFNUpdateRollback -StackName "myStack"
```

- Untuk detail API, lihat [ContinueUpdateRollback](#) di Referensi Alat AWS untuk PowerShell Cmdlet (V4).

Alat untuk PowerShell V5

Contoh 1: Melanjutkan rollback dari tumpukan bernama, yang seharusnya dalam status 'UPDATE_ROLLBACK_FAILED'. Jika rollback lanjutan berhasil, tumpukan akan memasukkan status 'UPDATE_ROLLBACK_COMPLETE'.

```
Resume-CFNUpdateRollback -StackName "myStack"
```

- Untuk detail API, lihat [ContinueUpdateRollback](#) di Referensi Alat AWS untuk PowerShell Cmdlet (V5).

Buat tumpukan

Gunakan [create-stack](#) perintah untuk membuat tumpukan. Anda harus memberikan nama tumpukan, lokasi templat yang valid, dan parameter masukan apa pun. Nama kunci parameter peka huruf besar/kecil. Jika Anda salah mengetik nama kunci parameter, CloudFormation tidak membuat tumpukan dan melaporkan bahwa template tidak berisi parameter itu.

Contoh berikut menunjukkan cara membuat tumpukan baru dengan nama, template, dan parameter input yang ditentukan.

CLI

AWS CLI

Untuk membuat AWS CloudFormation tumpukan

`create-stacks` Perintah berikut membuat tumpukan dengan nama `myteststack` menggunakan `sampletemplate.json` template:

```
aws cloudformation create-stack --stack-name myteststack --template-body file://sampletemplate.json --parameters ParameterKey=KeyPairName,ParameterValue=TestKey
ParameterKey=SubnetIDs,ParameterValue=SubnetID1\\,SubnetID2
```

Output:

```
{
  "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896"
}
```

Untuk informasi selengkapnya, lihat Tumpukan di Panduan AWS CloudFormation Pengguna.

- Untuk detail API, lihat [CreateStack](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell V4

Contoh 1: Membuat tumpukan baru dengan nama yang ditentukan. Template diurai dari konten yang disediakan dengan parameter kustomisasi ('PK1' dan 'PK2' mewakili nama parameter yang dideklarasikan dalam konten template, 'PV1' dan 'PV2' mewakili nilai untuk parameter tersebut. Parameter kustomisasi juga dapat ditentukan menggunakan 'Kunci' dan 'Nilai' alih-alih 'ParameterKey' dan 'ParameterValue'. Jika pembuatan tumpukan gagal, itu tidak akan digulung kembali.

```
New-CFNStack -StackName "myStack" `
              -TemplateBody "{TEMPLATE CONTENT HERE}" `
              -Parameter @( @{ ParameterKey="PK1"; ParameterValue="PV1" },
                             @{ ParameterKey="PK2"; ParameterValue="PV2" }) `
              -DisableRollback $true
```

Contoh 2: Membuat tumpukan baru dengan nama yang ditentukan. Template diurai dari konten yang disediakan dengan parameter kustomisasi ('PK1' dan 'PK2' mewakili nama parameter yang dideklarasikan dalam konten template, 'PV1' dan 'PV2' mewakili nilai untuk parameter tersebut. Parameter kustomisasi juga dapat ditentukan menggunakan 'Kunci' dan

'Nilai' alih-alih 'ParameterKey' dan 'ParameterValue'. Jika pembuatan tumpukan gagal, itu akan digulung kembali.

```
$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"
```

Contoh 3: Membuat tumpukan baru dengan nama yang ditentukan. Template diperoleh dari URL Amazon S3 dengan parameter kustomisasi ('PK1' mewakili nama parameter yang dideklarasikan dalam konten template, 'PV1' mewakili nilai untuk parameter. Parameter kustomisasi juga dapat ditentukan menggunakan 'Kunci' dan 'Nilai' alih-alih 'ParameterKey' dan 'ParameterValue'. Jika pembuatan tumpukan gagal, itu akan digulung kembali (sama seperti menentukan - DisableRollback \$false).

```
New-CFNStack -StackName "myStack" `
    -TemplateURL https://s3.amazonaws.com/amzn-s3-demo-bucket/
templatefile.template `
    -Parameter @{ ParameterKey="PK1"; ParameterValue="PV1" }
```

Contoh 4: Membuat tumpukan baru dengan nama yang ditentukan. Template diperoleh dari URL Amazon S3 dengan parameter kustomisasi ('PK1' mewakili nama parameter yang dideklarasikan dalam konten template, 'PV1' mewakili nilai untuk parameter. Parameter kustomisasi juga dapat ditentukan menggunakan 'Kunci' dan 'Nilai' alih-alih 'ParameterKey' dan 'ParameterValue'. Jika pembuatan tumpukan gagal, itu akan digulung kembali (sama seperti menentukan - DisableRollback \$false). Pemberitahuan yang ditentukan AENs akan menerima acara terkait tumpukan yang dipublikasikan.

```
New-CFNStack -StackName "myStack" `
    -TemplateURL https://s3.amazonaws.com/amzn-s3-demo-bucket/
templatefile.template `
    -Parameter @{ ParameterKey="PK1"; ParameterValue="PV1" } `
```

```
-NotificationARN @( "arn1", "arn2" )
```

- Untuk detail API, lihat [CreateStack](#) di Referensi Alat AWS untuk PowerShell Cmdlet (V4).

Alat untuk PowerShell V5

Contoh 1: Membuat tumpukan baru dengan nama yang ditentukan. Template diurai dari konten yang disediakan dengan parameter kustomisasi ('PK1' dan 'PK2' mewakili nama parameter yang dideklarasikan dalam konten template, 'PV1' dan 'PV2' mewakili nilai untuk parameter tersebut. Parameter kustomisasi juga dapat ditentukan menggunakan 'Kunci' dan 'Nilai' alih-alih 'ParameterKey' dan 'ParameterValue'. Jika pembuatan tumpukan gagal, itu tidak akan digulung kembali.

```
New-CFNStack -StackName "myStack" `
    -TemplateBody "{TEMPLATE CONTENT HERE}" `
    -Parameter @( @{ ParameterKey="PK1"; ParameterValue="PV1" },
    @{ ParameterKey="PK2"; ParameterValue="PV2" } ) `
    -DisableRollback $true
```

Contoh 2: Membuat tumpukan baru dengan nama yang ditentukan. Template diurai dari konten yang disediakan dengan parameter kustomisasi ('PK1' dan 'PK2' mewakili nama parameter yang dideklarasikan dalam konten template, 'PV1' dan 'PV2' mewakili nilai untuk parameter tersebut. Parameter kustomisasi juga dapat ditentukan menggunakan 'Kunci' dan 'Nilai' alih-alih 'ParameterKey' dan 'ParameterValue'. Jika pembuatan tumpukan gagal, itu akan digulung kembali.

```
$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"
```

Contoh 3: Membuat tumpukan baru dengan nama yang ditentukan. Template diperoleh dari URL Amazon S3 dengan parameter kustomisasi ('PK1' mewakili nama parameter yang

dideklarasikan dalam konten template, 'PV1' mewakili nilai untuk parameter. Parameter kustomisasi juga dapat ditentukan menggunakan 'Kunci' dan 'Nilai' alih-alih 'ParameterKey' dan 'ParameterValue'. Jika pembuatan tumpukan gagal, itu akan digulung kembali (sama seperti menentukan - DisableRollback \$false).

```
New-CFNStack -StackName "myStack" `
              -TemplateURL https://s3.amazonaws.com/amzn-s3-demo-bucket/
templatefile.template `
              -Parameter @{ ParameterKey="PK1"; ParameterValue="PV1" }
```

Contoh 4: Membuat tumpukan baru dengan nama yang ditentukan. Template diperoleh dari URL Amazon S3 dengan parameter kustomisasi ('PK1' mewakili nama parameter yang dideklarasikan dalam konten template, 'PV1' mewakili nilai untuk parameter. Parameter kustomisasi juga dapat ditentukan menggunakan 'Kunci' dan 'Nilai' alih-alih 'ParameterKey' dan 'ParameterValue'. Jika pembuatan tumpukan gagal, itu akan digulung kembali (sama seperti menentukan - DisableRollback \$false). Pemberitahuan yang ditentukan AENs akan menerima acara terkait tumpukan yang dipublikasikan.

```
New-CFNStack -StackName "myStack" `
              -TemplateURL https://s3.amazonaws.com/amzn-s3-demo-bucket/
templatefile.template `
              -Parameter @{ ParameterKey="PK1"; ParameterValue="PV1" } `
              -NotificationARN @( "arn1", "arn2" )
```

- Untuk detail API, lihat [CreateStack](#) di Referensi Alat AWS untuk PowerShell Cmdlet (V5).

Note

Anda dapat menggunakan AWS CLI `--template-url` opsi untuk menentukan lokasi file template di Amazon S3 atau AWS Systems Manager. Untuk Amazon S3, URL harus dimulai dengan `https://`. Situs web statis S3 URLs tidak didukung.

```
--template-url https://s3.region-code.amazonaws.com/bucket-name/template-name
```

Untuk AWS Systems Manager, gunakan format berikut:

```
--template-url "ssm-doc://arn:aws:ssm:region-code:account-id:document/document-name"
```

Buat tumpukan yang mencakup transformasi

Gunakan [deploy](#) perintah untuk membuat tumpukan yang mencakup transformasi. Saat Anda membuat tumpukan dari template yang menyertakan transformasi, Anda harus menggunakan set perubahan. `deploy` Perintah menggabungkan dua langkah (membuat set perubahan dan menjalankannya) menjadi satu perintah.

AWS CLI

`deploy` Perintah berikut membuat tumpukan dengan nama, template, dan parameter input yang ditentukan.

```
aws cloudformation deploy --stack-name myteststack \  
  --template /path_to_template/my-template.json \  
  --parameter-overrides Key1=Value1 Key2=Value2
```

Hapus tumpukan

Gunakan [delete-stack](#) perintah untuk menghapus tumpukan. Untuk informasi selengkapnya, lihat [Hapus tumpukan](#).

CLI

AWS CLI

Untuk menghapus tumpukan

`delete-stack` Contoh berikut menghapus tumpukan yang ditentukan.

```
aws cloudformation delete-stack \  
  --stack-name my-stack
```

Perintah ini tidak menghasilkan output.

- Untuk detail API, lihat [DeleteStack](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell V4

Contoh 1: Menghapus tumpukan yang ditentukan.

```
Remove-CFNStack -StackName "myStack"
```

- Untuk detail API, lihat [DeleteStack](#) di Referensi Alat AWS untuk PowerShell Cmdlet (V4).

Alat untuk PowerShell V5

Contoh 1: Menghapus tumpukan yang ditentukan.

```
Remove-CFNStack -StackName "myStack"
```

- Untuk detail API, lihat [DeleteStack](#) di Referensi Alat AWS untuk PowerShell Cmdlet (V5).

Jika penghapusan gagal dan mengembalikan DELETE_FAILED status, Anda dapat memilih untuk menghapus tumpukan dengan paksa menggunakan opsi. --deletion-mode Ini adalah nilai-nilai berikut yang dapat digunakan dengan `deletion-mode`:

- STANDARD: Menghapus tumpukan secara normal. Ini adalah mode penghapusan default.
- FORCE_DELETE_STACK: Menghapus tumpukan dan melewati semua sumber daya yang gagal dihapus.

AWS CLI

`delete-stack` Command force berikut menghapus *myteststack* tumpukan menggunakan FORCE_DELETE_STACK nilai dengan `deletion-mode` parameter:

```
aws cloudformation delete-stack --stack-name myteststack \  
--deletion-mode FORCE_DELETE_STACK
```

Perintah ini tidak menghasilkan output.

Setelah menggunakan FORCE_DELETE_STACK, Anda dapat menggunakan `list-stack-resources` perintah untuk membuat daftar sumber daya yang dilewati selama proses penghapusan

tumpukan. Sumber daya yang disimpan akan menampilkan status DELETE_SKIPPED. Untuk informasi selengkapnya, lihat [Daftar sumber daya tumpukan](#).

Jelaskan peristiwa tumpukan

Gunakan [describe-stack-events](#) perintah untuk menggambarkan peristiwa tumpukan. Untuk informasi selengkapnya, lihat [Pantau kemajuan tumpukan](#).

CLI

AWS CLI

Untuk menggambarkan peristiwa tumpukan

`describe-stack-events` Contoh berikut menampilkan 2 peristiwa terbaru untuk tumpukan tertentu.

```
aws cloudformation describe-stack-events \
  --stack-name my-stack \
  --max-items 2

{
  "StackEvents": [
    {
      "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
      "EventId": "4e1516d0-e4d6-xmpl-b94f-0a51958a168c",
      "StackName": "my-stack",
      "LogicalResourceId": "my-stack",
      "PhysicalResourceId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
      "ResourceType": "AWS::CloudFormation::Stack",
      "Timestamp": "2019-10-02T05:34:29.556Z",
      "ResourceStatus": "UPDATE_COMPLETE"
    },
    {
      "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
      "EventId": "4dd3c810-e4d6-xmpl-bade-0aaf8b31ab7a",
      "StackName": "my-stack",
      "LogicalResourceId": "my-stack",
      "PhysicalResourceId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
```

```

        "ResourceType": "AWS::CloudFormation::Stack",
        "Timestamp": "2019-10-02T05:34:29.127Z",
        "ResourceStatus": "UPDATE_COMPLETE_CLEANUP_IN_PROGRESS"
    },
    "NextToken": "eyJ0ZXh0VG9XMPLiOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAyfQ=="
}

```

- Untuk detail API, lihat [DescribeStackEvents](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell V4

Contoh 1: Mengembalikan semua peristiwa terkait tumpukan untuk tumpukan tertentu.

```
Get-CFNStackEvent -StackName "myStack"
```

- Untuk detail API, lihat [DescribeStackEvents](#) di Referensi Alat AWS untuk PowerShell Cmdlet (V4).

Alat untuk PowerShell V5

Contoh 1: Mengembalikan semua peristiwa terkait tumpukan untuk tumpukan tertentu.

```
Get-CFNStackEvent -StackName "myStack"
```

- Untuk detail API, lihat [DescribeStackEvents](#) di Referensi Alat AWS untuk PowerShell Cmdlet (V5).

Jelaskan sumber daya tumpukan

CLI

AWS CLI

Untuk mendapatkan informasi tentang sumber daya tumpukan

`describe-stack-resource` Contoh berikut menampilkan rincian untuk sumber daya bernama MyFunction dalam tumpukan tertentu.

```
aws cloudformation describe-stack-resource \  
  --stack-name MyStack \  
  --logical-resource-id MyFunction
```

Output:

```
{  
  "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"  
    }  
  }  
}
```

- Untuk detail API, lihat [DescribeStackResource](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell V4

Contoh 1: Mengembalikan deskripsi sumber daya yang diidentifikasi dalam template yang terkait dengan tumpukan yang ditentukan oleh ID logis DBInstance “Saya”.

```
Get-CFNStackResource -StackName "myStack" -LogicalResourceId "MyDBInstance"
```

- Untuk detail API, lihat [DescribeStackResource](#) di Referensi Alat AWS untuk PowerShell Cmdlet (V4).

Alat untuk PowerShell V5

Contoh 1: Mengembalikan deskripsi sumber daya yang diidentifikasi dalam template yang terkait dengan tumpukan yang ditentukan oleh ID logis DBInstance “Saya”.

```
Get-CFNStackResource -StackName "myStack" -LogicalResourceId "MyDBInstance"
```

- Untuk detail API, lihat [DescribeStackResource](#) di Referensi Alat AWS untuk PowerShell Cmdlet (V5).

Jelaskan sumber daya tumpukan

CLI

AWS CLI

Untuk mendapatkan informasi tentang sumber daya tumpukan

`describe-stack-resources` Contoh berikut menampilkan rincian untuk sumber daya dalam tumpukan tertentu.

```
aws cloudformation describe-stack-resources \
  --stack-name my-stack
```

Output:

```
{
  "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"
        }
    }
]
}

```

- Untuk detail API, lihat [DescribeStackResources](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell V4

Contoh 1: Mengembalikan deskripsi AWS sumber daya hingga 100 sumber daya yang terkait dengan tumpukan yang ditentukan. Untuk mendapatkan rincian semua sumber daya yang terkait dengan tumpukan gunakan `Get-CFNStackResourceSummary`, yang juga mendukung paging manual hasil.

```
Get-CFNStackResourceList -StackName "myStack"
```

Contoh 2: Mengembalikan deskripsi EC2 instance Amazon yang diidentifikasi dalam template yang terkait dengan tumpukan yang ditentukan oleh ID logis "Ec2Instance".

```
Get-CFNStackResourceList -StackName "myStack" -LogicalResourceId "Ec2Instance"
```

Contoh 3: Mengembalikan deskripsi hingga 100 sumber daya yang terkait dengan tumpukan yang berisi instance Amazon yang diidentifikasi oleh ID EC2 instance "i-123456". Untuk mendapatkan rincian semua sumber daya yang terkait dengan tumpukan gunakan `Get-CFNStackResourceSummary`, yang juga mendukung paging manual hasil.

```
Get-CFNStackResourceList -PhysicalResourceId "i-123456"
```

Contoh 4: Mengembalikan deskripsi EC2 instance Amazon yang diidentifikasi oleh ID logis "Ec2Instance" dalam template untuk tumpukan. Tumpukan diidentifikasi menggunakan ID sumber daya fisik dari sumber daya yang dikandungnya, dalam hal ini juga instance Amazon dengan ID EC2 instance "i-123456". Sumber daya fisik yang berbeda juga dapat digunakan untuk mengidentifikasi tumpukan tergantung pada konten template, misalnya bucket Amazon S3.

```
Get-CFNStackResourceList -PhysicalResourceId "i-123456" -LogicalResourceId "Ec2Instance"
```

- Untuk detail API, lihat [DescribeStackResources](#) di Referensi Alat AWS untuk PowerShell Cmdlet (V4).

Alat untuk PowerShell V5

Contoh 1: Mengembalikan deskripsi AWS sumber daya hingga 100 sumber daya yang terkait dengan tumpukan yang ditentukan. Untuk mendapatkan rincian semua sumber daya yang terkait dengan tumpukan gunakan `Get-CFNStackResourceSummary`, yang juga mendukung paging manual hasil.

```
Get-CFNStackResourceList -StackName "myStack"
```

Contoh 2: Mengembalikan deskripsi EC2 instance Amazon yang diidentifikasi dalam template yang terkait dengan tumpukan yang ditentukan oleh ID logis "Ec2Instance".

```
Get-CFNStackResourceList -StackName "myStack" -LogicalResourceId "Ec2Instance"
```

Contoh 3: Mengembalikan deskripsi hingga 100 sumber daya yang terkait dengan tumpukan yang berisi instance Amazon yang diidentifikasi oleh ID EC2 instance "i-123456". Untuk mendapatkan rincian semua sumber daya yang terkait dengan tumpukan gunakan `Get-CFNStackResourceSummary`, yang juga mendukung paging manual hasil.

```
Get-CFNStackResourceList -PhysicalResourceId "i-123456"
```

Contoh 4: Mengembalikan deskripsi EC2 instance Amazon yang diidentifikasi oleh ID logis "Ec2Instance" dalam template untuk tumpukan. Tumpukan diidentifikasi menggunakan ID sumber daya fisik dari sumber daya yang dikandungnya, dalam hal ini juga instance Amazon dengan ID EC2 instance "i-123456". Sumber daya fisik yang berbeda juga dapat digunakan untuk mengidentifikasi tumpukan tergantung pada konten template, misalnya bucket Amazon S3.

```
Get-CFNStackResourceList -PhysicalResourceId "i-123456" -LogicalResourceId  
"Ec2Instance"
```

- Untuk detail API, lihat [DescribeStackResources](#) di Referensi Alat AWS untuk PowerShell Cmdlet (V5).

Jelaskan tumpukan

CLI

AWS CLI

Untuk menggambarkan AWS CloudFormation tumpukan

`describe-stacks` Perintah berikut menunjukkan informasi ringkasan untuk `myteststack` tumpukan:

```
aws cloudformation describe-stacks --stack-name myteststack
```

Output:

```
{  
  "Stacks": [  
    {  
      "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/  
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896",  
      "Description": "AWS CloudFormation Sample Template S3_Bucket: Sample  
template showing how to create a publicly accessible S3 bucket. **WARNING** This  
template creates an S3 bucket. You will be billed for the AWS resources used if  
you create a stack from this template.",  
      "Tags": [],  
    }  
  ]  
}
```

```
        "Outputs": [
            {
                "Description": "Name of S3 bucket to hold website content",
                "OutputKey": "BucketName",
                "OutputValue": "myteststack-s3bucket-jssofi1zie2w"
            }
        ],
        "StackStatusReason": null,
        "CreationTime": "2013-08-23T01:02:15.422Z",
        "Capabilities": [],
        "StackName": "myteststack",
        "StackStatus": "CREATE_COMPLETE",
        "DisableRollback": false
    }
}
```

Untuk informasi selengkapnya, lihat Tumpukan di Panduan AWS CloudFormation Pengguna.

- Untuk detail API, lihat [DescribeStacks](#) di Referensi AWS CLI Perintah.

Go

SDK untuk Go V2

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
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
}
```

- Untuk detail API, lihat [DescribeStacks](#) di Referensi AWS SDK untuk Go API.

PowerShell

Alat untuk PowerShell V4

Contoh 1: Mengembalikan koleksi instance Stack yang menjelaskan semua tumpukan pengguna.

```
Get-CFNStack
```

Contoh 2: Mengembalikan instance Stack yang menjelaskan tumpukan yang ditentukan

```
Get-CFNStack -StackName "myStack"
```

- Untuk detail API, lihat [DescribeStacks](#) di Referensi Alat AWS untuk PowerShell Cmdlet (V4).

Alat untuk PowerShell V5

Contoh 1: Mengembalikan koleksi instance Stack yang menjelaskan semua tumpukan pengguna.

```
Get-CFNStack
```

Contoh 2: Mengembalikan instance Stack yang menjelaskan tumpukan yang ditentukan

```
Get-CFNStack -StackName "myStack"
```

- Untuk detail API, lihat [DescribeStacks](#) di Referensi Alat AWS untuk PowerShell Cmdlet (V5).

Secara default, `describe-stacks` perintah mengembalikan nilai parameter. Untuk mencegah nilai parameter sensitif seperti kata sandi dikembalikan, sertakan `NoEcho` properti yang disetel ke `TRUE` dalam CloudFormation templat Anda.

Important

Menggunakan `NoEcho` atribut tidak menutupi informasi apa pun yang disimpan sebagai berikut:

- Bagian Metadata template. CloudFormation tidak mengubah, memodifikasi, atau menyunting informasi apa pun yang Anda sertakan di Metadata bagian ini. Untuk informasi selengkapnya, lihat [Metadata](#).
- Bagian Outputs template. Untuk informasi selengkapnya, lihat [Output](#).
- MetadataAtribut definisi sumber daya. Untuk informasi lebih lanjut, lihat [Atribut Metadata](#).

Kami sangat menyarankan Anda untuk tidak menggunakan mekanisme ini untuk memasukkan informasi sensitif, seperti kata sandi atau rahasia.

Important

Daripada menyematkan informasi sensitif secara langsung di CloudFormation template Anda, kami sarankan Anda menggunakan parameter dinamis dalam template tumpukan untuk

merefereksikan informasi sensitif yang disimpan dan dikelola di luar CloudFormation, seperti di AWS Systems Manager Parameter Store atau AWS Secrets Manager.

Untuk informasi selengkapnya, lihat [Do not embed credentials in your templates](#) best practice.

Dapatkan template

CLI

AWS CLI

Untuk melihat badan template untuk AWS CloudFormation tumpukan

get-templatePerintah berikut menunjukkan template untuk myteststack tumpukan:

```
aws cloudformation get-template --stack-name myteststack
```

Output:

```
{
  "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"
        }
      }
    }
  }
}
```

```
}  
}  
}
```

- Untuk detail API, lihat [GetTemplate](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell V4

Contoh 1: Mengembalikan template yang terkait dengan tumpukan tertentu.

```
Get-CFNTemplate -StackName "myStack"
```

- Untuk detail API, lihat [GetTemplate](#) di Referensi Alat AWS untuk PowerShell Cmdlet (V4).

Alat untuk PowerShell V5

Contoh 1: Mengembalikan template yang terkait dengan tumpukan tertentu.

```
Get-CFNTemplate -StackName "myStack"
```

- Untuk detail API, lihat [GetTemplate](#) di Referensi Alat AWS untuk PowerShell Cmdlet (V5).

Daftar sumber daya tumpukan

CLI

AWS CLI

Untuk membuat daftar sumber daya dalam tumpukan

Perintah berikut menampilkan daftar sumber daya dalam tumpukan yang ditentukan.

```
aws cloudformation list-stack-resources \  
  --stack-name my-stack
```

Output:

```
{
```

```

"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-HIZXMPLE0M9E",
    "ResourceType": "AWS::IAM::Role",
    "LastUpdatedTimestamp": "2019-10-02T04:34:06.350Z",
    "ResourceStatus": "CREATE_COMPLETE",
    "DriftInformation": {
      "StackResourceDriftStatus": "IN_SYNC"
    }
  }
]
}

```

- Untuk detail API, lihat [ListStackResources](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell V4

Contoh 1: Mengembalikan deskripsi dari semua sumber daya yang terkait dengan tumpukan tertentu.

```
Get-CFNStackResourceSummary -StackName "myStack"
```

- Untuk detail API, lihat [ListStackResources](#) di Referensi Alat AWS untuk PowerShell Cmdlet (V4).

Alat untuk PowerShell V5

Contoh 1: Mengembalikan deskripsi dari semua sumber daya yang terkait dengan tumpukan tertentu.

```
Get-CFNStackResourceSummary -StackName "myStack"
```

- Untuk detail API, lihat [ListStackResources](#) di Referensi Alat AWS untuk PowerShell Cmdlet (V5).

Daftar tumpukan

Gunakan [list-stacks](#) perintah untuk membuat daftar tumpukan. Untuk daftar hanya tumpukan dengan kode status yang ditentukan, sertakan `--stack-status-filter` opsi. Anda dapat menentukan satu atau lebih kode status tumpukan untuk `--stack-status-filter` opsi tersebut. Untuk informasi selengkapnya, lihat [Kode status tumpukan](#).

CLI

AWS CLI

Untuk daftar AWS CloudFormation tumpukan

`list-stacks` Perintah berikut menunjukkan ringkasan semua tumpukan yang memiliki status `CREATE_COMPLETE`:

```
aws cloudformation list-stacks --stack-status-filter CREATE_COMPLETE
```

Output:

```
[
  {
    "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"
  }
]
```

- Untuk detail API, lihat [ListStacks](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell V4

Contoh 1: Mengembalikan informasi ringkasan untuk semua tumpukan.

```
Get-CFNStackSummary
```

Contoh 2: Mengembalikan informasi ringkasan untuk semua tumpukan yang sedang dibuat.

```
Get-CFNStackSummary -StackStatusFilter "CREATE_IN_PROGRESS"
```

Contoh 3: Mengembalikan informasi ringkasan untuk semua tumpukan yang sedang dibuat atau diperbarui.

```
Get-CFNStackSummary -StackStatusFilter @("CREATE_IN_PROGRESS",
"UPDATE_IN_PROGRESS")
```

- Untuk detail API, lihat [ListStacks](#) di Referensi Alat AWS untuk PowerShell Cmdlet (V4).

Alat untuk PowerShell V5

Contoh 1: Mengembalikan informasi ringkasan untuk semua tumpukan.

```
Get-CFNStackSummary
```

Contoh 2: Mengembalikan informasi ringkasan untuk semua tumpukan yang sedang dibuat.

```
Get-CFNStackSummary -StackStatusFilter "CREATE_IN_PROGRESS"
```

Contoh 3: Mengembalikan informasi ringkasan untuk semua tumpukan yang sedang dibuat atau diperbarui.

```
Get-CFNStackSummary -StackStatusFilter @("CREATE_IN_PROGRESS",  
"UPDATE_IN_PROGRESS")
```

- Untuk detail API, lihat [ListStacks](#) di Referensi Alat AWS untuk PowerShell Cmdlet (V5).

Perbarui tumpukan

Gunakan perintah [update-stack](#) untuk memperbarui tumpukan secara langsung. Anda menentukan tumpukan, dan parameter nilai dan kemampuan yang akan diperbarui, dan, jika Anda ingin menggunakan templat yang diperbarui, nama templat. Untuk informasi selengkapnya, lihat [Perbarui tumpukan secara langsung](#).

CLI

AWS CLI

Untuk memperbarui AWS CloudFormation tumpukan

update-stack Perintah berikut memperbarui template dan parameter input untuk mystack tumpukan:

```
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
```

update-stack Perintah berikut memperbarui hanya nilai SubnetIDs parameter untuk mystack tumpukan. Jika Anda tidak menentukan nilai parameter, nilai default yang ditentukan dalam template digunakan:

```
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
```

update-stack Perintah berikut menambahkan dua topik notifikasi mystack tumpukan ke tumpukan:


```
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"
```

Untuk informasi selengkapnya, lihat [AWS CloudFormation tumpukan pembaruan](#) di Panduan AWS CloudFormation Pengguna.

- Untuk detail API, lihat [UpdateStack](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell V4

Contoh 1: Memperbarui tumpukan 'MyStack' dengan template dan parameter kustomisasi yang ditentukan. 'PK1' mewakili nama parameter yang dideklarasikan dalam template dan 'PV1' mewakili nilainya. Parameter kustomisasi juga dapat ditentukan menggunakan 'Kunci' dan 'Nilai' alih-alih 'ParameterKey' dan 'ParameterValue'.

```
Update-CFNStack -StackName "myStack" `
    -TemplateBody "{Template Content Here}" `
    -Parameter @{ ParameterKey="PK1"; ParameterValue="PV1" }
```

Contoh 2: Memperbarui tumpukan 'MyStack' dengan template dan parameter kustomisasi yang ditentukan. 'PK1' dan 'PK2' mewakili nama parameter yang dideklarasikan dalam templat, 'PV1' dan 'PV2' mewakili nilai yang diminta. Parameter kustomisasi juga dapat ditentukan menggunakan 'Kunci' dan 'Nilai' alih-alih 'ParameterKey' dan 'ParameterValue'.

```
Update-CFNStack -StackName "myStack" `
    -TemplateBody "{Template Content Here}" `
    -Parameter @( @{ ParameterKey="PK1"; ParameterValue="PV1" },
    @{ ParameterKey="PK2"; ParameterValue="PV2" } )
```

Contoh 3: Memperbarui tumpukan 'MyStack' dengan template dan parameter kustomisasi yang ditentukan. 'PK1' mewakili nama parameter yang dideklarasikan dalam template dan 'PV2' mewakili nilainya. Parameter kustomisasi juga dapat ditentukan menggunakan 'Kunci' dan 'Nilai' alih-alih 'ParameterKey' dan 'ParameterValue'.

```
Update-CFNStack -StackName "myStack" -TemplateBody "{Template Content Here}" -Parameters @{ ParameterKey="PK1"; ParameterValue="PV1" }
```

Contoh 4: Memperbarui tumpukan 'MyStack' dengan templat yang ditentukan, diperoleh dari Amazon S3, dan parameter penyesuaian. 'PK1' dan 'PK2' mewakili nama parameter yang dideklarasikan dalam templat, 'PV1' dan 'PV2' mewakili nilai yang diminta. Parameter kustomisasi juga dapat ditentukan menggunakan 'Kunci' dan 'Nilai' alih-alih 'ParameterKey' dan 'ParameterValue'.

```
Update-CFNStack -StackName "myStack" `
                -TemplateURL https://s3.amazonaws.com/amzn-s3-demo-bucket/
templatefile.template `
                -Parameter @( @{ ParameterKey="PK1"; ParameterValue="PV1" },
@{ ParameterKey="PK2"; ParameterValue="PV2" } )
```

Contoh 5: Memperbarui tumpukan 'MyStack', yang diasumsikan dalam contoh ini berisi sumber daya IAM, dengan templat yang ditentukan, diperoleh dari Amazon S3, dan parameter penyesuaian. 'PK1' dan 'PK2' mewakili nama parameter yang dideklarasikan dalam templat, 'PV1' dan 'PV2' mewakili nilai yang diminta. Parameter kustomisasi juga dapat ditentukan menggunakan 'Kunci' dan 'Nilai' alih-alih 'ParameterKey' dan 'ParameterValue'. Tumpukan yang berisi sumber daya IAM mengharuskan Anda untuk menentukan parameter -Capabilities "CAPABILITY_IAM" jika tidak pembaruan akan gagal dengan kesalahan "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"
```

- Untuk detail API, lihat [UpdateStack](#) di Referensi Alat AWS untuk PowerShell Cmdlet (V4).

Alat untuk PowerShell V5

Contoh 1: Memperbarui tumpukan 'MyStack' dengan template dan parameter kustomisasi yang ditentukan. 'PK1' mewakili nama parameter yang dideklarasikan dalam template dan 'PV1' mewakili nilainya. Parameter kustomisasi juga dapat ditentukan menggunakan 'Kunci' dan 'Nilai' alih-alih 'ParameterKey' dan 'ParameterValue'.

```
Update-CFNStack -StackName "myStack" `
                -TemplateBody "{Template Content Here}" `
                -Parameter @{ ParameterKey="PK1"; ParameterValue="PV1" }
```

Contoh 2: Memperbarui tumpukan 'MyStack' dengan template dan parameter kustomisasi yang ditentukan. 'PK1' dan 'PK2' mewakili nama parameter yang dideklarasikan dalam templat, 'PV1' dan 'PV2' mewakili nilai yang diminta. Parameter kustomisasi juga dapat ditentukan menggunakan 'Kunci' dan 'Nilai' alih-alih 'ParameterKey' dan 'ParameterValue'.

```
Update-CFNStack -StackName "myStack" `
    -TemplateBody "{Template Content Here}" `
    -Parameter @( @{ ParameterKey="PK1"; ParameterValue="PV1" },
    @{ ParameterKey="PK2"; ParameterValue="PV2" } )
```

Contoh 3: Memperbarui tumpukan 'MyStack' dengan template dan parameter kustomisasi yang ditentukan. 'PK1' mewakili nama parameter yang dideklarasikan dalam template dan 'PV2' mewakili nilainya. Parameter kustomisasi juga dapat ditentukan menggunakan 'Kunci' dan 'Nilai' alih-alih 'ParameterKey' dan 'ParameterValue'.

```
Update-CFNStack -StackName "myStack" -TemplateBody "{Template Content Here}" -
Parameters @{ ParameterKey="PK1"; ParameterValue="PV1" }
```

Contoh 4: Memperbarui tumpukan 'MyStack' dengan templat yang ditentukan, diperoleh dari Amazon S3, dan parameter penyesuaian. 'PK1' dan 'PK2' mewakili nama parameter yang dideklarasikan dalam templat, 'PV1' dan 'PV2' mewakili nilai yang diminta. Parameter kustomisasi juga dapat ditentukan menggunakan 'Kunci' dan 'Nilai' alih-alih 'ParameterKey' dan 'ParameterValue'.

```
Update-CFNStack -StackName "myStack" `
    -TemplateURL https://s3.amazonaws.com/amzn-s3-demo-bucket/
    templatefile.template `
    -Parameter @( @{ ParameterKey="PK1"; ParameterValue="PV1" },
    @{ ParameterKey="PK2"; ParameterValue="PV2" } )
```

Contoh 5: Memperbarui tumpukan 'MyStack', yang diasumsikan dalam contoh ini berisi sumber daya IAM, dengan templat yang ditentukan, diperoleh dari Amazon S3, dan parameter penyesuaian. 'PK1' dan 'PK2' mewakili nama parameter yang dideklarasikan dalam templat, 'PV1' dan 'PV2' mewakili nilai yang diminta. Parameter kustomisasi juga dapat ditentukan menggunakan 'Kunci' dan 'Nilai' alih-alih 'ParameterKey' dan 'ParameterValue'. Tumpukan yang berisi sumber daya IAM mengharuskan Anda untuk menentukan parameter -Capabilities "CAPABILITY_IAM" jika tidak pembaruan akan gagal dengan kesalahan "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"
```

- Untuk detail API, lihat [UpdateStack](#) di Referensi Alat AWS untuk PowerShell Cmdlet (V5).

Note

Untuk menghapus semua notifikasi, tentukan [] untuk --notification-arns opsi tersebut.

Validasi template Anda

Gunakan [validate-template](#) perintah untuk memeriksa file template Anda untuk kesalahan sintaks.

Selama validasi, periksa CloudFormation terlebih dahulu apakah template adalah JSON yang valid. Jika bukan, CloudFormation memeriksa apakah templat YAML valid. Jika kedua pemeriksaan gagal, CloudFormation mengembalikan kesalahan validasi template.

CLI

AWS CLI

Untuk memvalidasi template AWS CloudFormation

`validate-template` Perintah berikut memvalidasi `sampletemplate.json` template:

```
aws cloudformation validate-template --template-body file://sampletemplate.json
```

Output:

```
{
  "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": []  
}
```

Untuk informasi selengkapnya, lihat Bekerja dengan AWS CloudFormation Template di Panduan AWS CloudFormation Pengguna.

- Untuk detail API, lihat [ValidateTemplate](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell V4

Contoh 1: Memvalidasi konten template yang ditentukan. Output merinci kemampuan, deskripsi, dan parameter template.

```
Test-CFNTemplate -TemplateBody "{TEMPLATE CONTENT HERE}"
```

Contoh 2: Memvalidasi template tertentu yang diakses melalui URL Amazon S3. Output merinci kemampuan, deskripsi, dan parameter template.

```
Test-CFNTemplate -TemplateURL https://s3.amazonaws.com/amzn-s3-demo-bucket/  
templatefile.template
```

- Untuk detail API, lihat [ValidateTemplate](#) di Referensi Alat AWS untuk PowerShell Cmdlet (V4).

Alat untuk PowerShell V5

Contoh 1: Memvalidasi konten template yang ditentukan. Output merinci kemampuan, deskripsi, dan parameter template.

```
Test-CFNTemplate -TemplateBody "{TEMPLATE CONTENT HERE}"
```

Contoh 2: Memvalidasi template tertentu yang diakses melalui URL Amazon S3. Output merinci kemampuan, deskripsi, dan parameter template.

```
Test-CFNTemplate -TemplateURL https://s3.amazonaws.com/amzn-s3-demo-bucket/  
templatefile.template
```

- Untuk detail API, lihat [ValidateTemplate](#) di Referensi Alat AWS untuk PowerShell Cmdlet (V5).

Berikut ini adalah contoh respon yang menghasilkan kesalahan validasi.

```
{
  "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

`validate-template` Perintah ini dirancang untuk memeriksa hanya sintaks template Anda. Itu tidak memastikan bahwa nilai properti yang telah Anda tentukan untuk sumber daya valid untuk sumber daya tersebut. Juga tidak menentukan jumlah sumber daya yang akan ada saat tumpukan dibuat.

Untuk memeriksa validitas operasional, Anda perlu mencoba membuat tumpukan. Tidak ada kotak pasir atau area pengujian untuk AWS CloudFormation tumpukan, jadi Anda dikenakan biaya untuk sumber daya yang Anda buat selama pengujian.

Unggah artefak lokal ke bucket S3 dengan AWS CLI

Anda dapat menggunakan AWS CLI untuk mengunggah artefak lokal yang direferensikan oleh CloudFormation template ke bucket Amazon S3. Artefak lokal adalah file yang Anda referensi dalam template Anda. Alih-alih mengunggah file secara manual ke bucket S3 dan kemudian

menambahkan lokasinya ke template Anda, Anda dapat menentukan artefak lokal di template Anda dan menggunakan [package](#) perintah untuk mengunggahnya dengan cepat.

Artefak lokal adalah jalur ke file atau folder yang diunggah package perintah ke Amazon S3. Misalnya, artefak dapat berupa jalur lokal ke kode sumber AWS Lambda fungsi Anda atau file OpenAPI Amazon API Gateway REST API.

Saat menggunakan package perintah:

- Jika Anda menentukan file, perintah secara langsung mengunggahnya ke bucket S3.
- Jika Anda menentukan folder, perintah membuat .zip file untuk folder, dan kemudian mengunggah .zip file.
- Jika Anda tidak menentukan jalur, perintah akan membuat .zip file untuk direktori kerja dan mengunggahnya.

Anda dapat menentukan jalur absolut atau relatif, dimana jalur relatif bersifat relatif untuk lokasi templat Anda.

Setelah mengunggah artifact, perintah menampilkan salinan templat Anda, mengganti referensi ke artifact lokal dengan lokasi S3 tempat perintah mengunggah artifact. Anda kemudian dapat menggunakan template yang dikembalikan untuk membuat atau memperbarui tumpukan.

Note

Anda dapat menggunakan artifact lokal untuk atribut sumber daya yang didukung perintah package saja. Untuk informasi selengkapnya tentang perintah ini dan daftar properti sumber daya yang didukung, lihat [package](#) dokumentasi di [Referensi AWS CLI Perintah](#).

Prasyarat

Sebelum memulai, Anda harus memiliki bucket Amazon S3 yang ada.

Package dan deploy template dengan artefak lokal

Templat berikut menentukan artifact lokal untuk kode sumber fungsi Lambda. Kode sumber disimpan di `/home/user/code/lambdafunction` folder.

Template asli

```
{
  "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"
      }
    }
  }
}
```

[package](#) Perintah berikut membuat dan mengunggah .zip file folder kode sumber fungsi ke root bucket yang ditentukan.

```
aws cloudformation package \
  --s3-bucket amzn-s3-demo-bucket \
  --template /path_to_template/template.json \
  --output-template-file packaged-template.json \
  --output json
```

Perintah menghasilkan template baru di jalur yang ditentukan oleh `--output-template-file`. Ini menggantikan referensi artefak dengan lokasi Amazon S3, seperti yang ditunjukkan di bawah ini. `.zipFile` diberi nama menggunakan MD5 checksum dari isi folder, daripada menggunakan nama folder itu sendiri.

Template yang dihasilkan

```
{
  "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"
    }
}
}
```

Setelah Anda mengemas artefak template Anda, gunakan template yang diproses menggunakan perintah. [deploy](#)

```
aws cloudformation deploy \  
  --template-file packaged-template.json \  
  --stack-name stack-name
```

Saat menerapkan template yang lebih besar dari 51.200 byte, gunakan deploy perintah dengan --s3-bucket opsi untuk mengunggah template Anda ke S3, seperti pada contoh berikut.

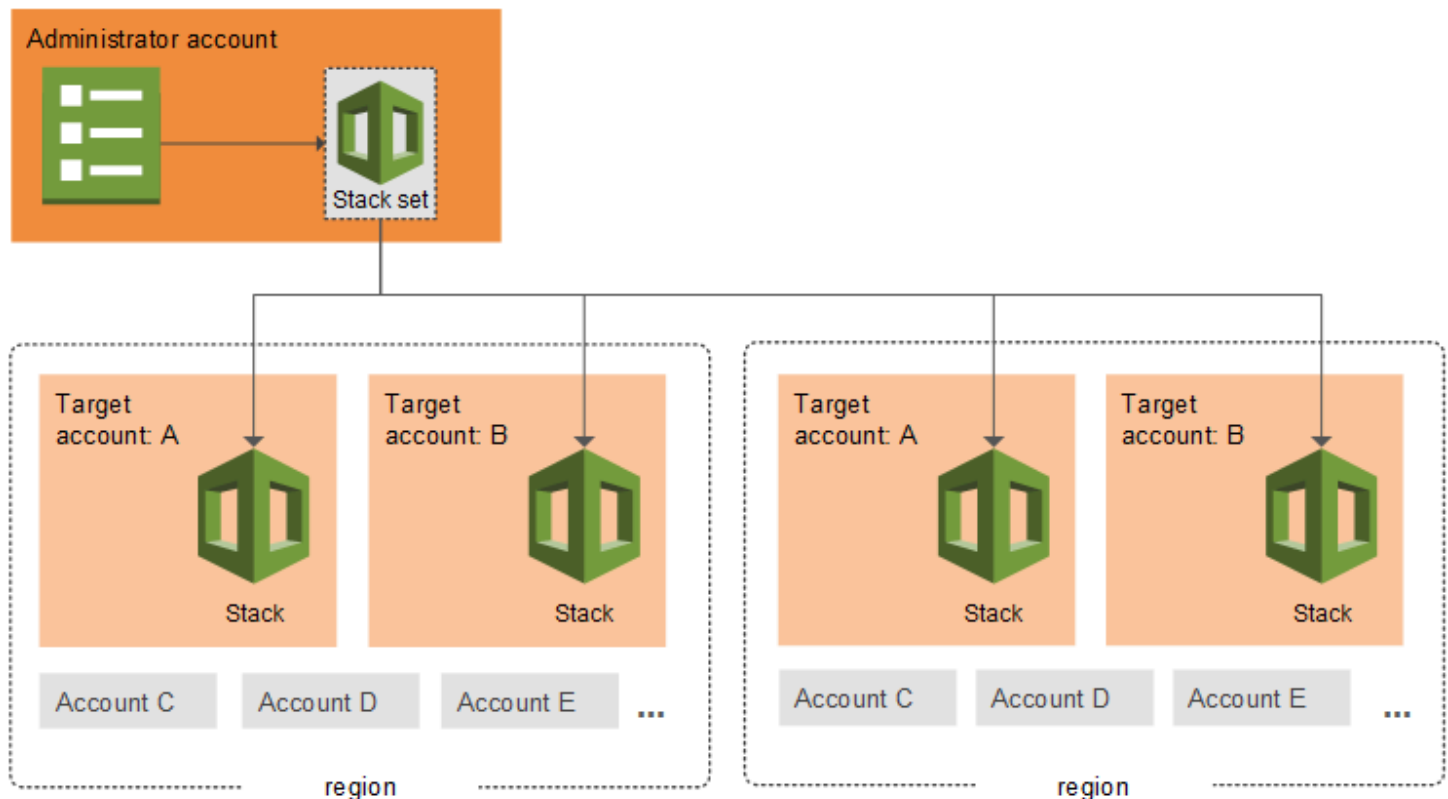
```
aws cloudformation deploy \  
  --template-file packaged-template.json \  
  --stack-name stack-name \  
  --s3-bucket amzn-s3-demo-bucket
```

Note

Untuk contoh lain menggunakan package perintah untuk mengunggah artefak lokal, lihat [Pisahkan template menjadi potongan-potongan yang dapat digunakan kembali menggunakan tumpukan bersarang](#).

Mengelola tumpukan di seluruh akun dan Wilayah dengan StackSets

AWS CloudFormation StackSets memperluas kemampuan tumpukan dengan memungkinkan Anda membuat, memperbarui, atau menghapus tumpukan di beberapa akun dan Wilayah AWS dengan satu operasi. Dengan menggunakan akun administrator, Anda menentukan dan mengelola CloudFormation templat, dan menggunakan templat sebagai dasar untuk menyediakan tumpukan ke akun target yang dipilih di seluruh yang ditentukan. Wilayah AWS



Bagian ini membantu Anda mulai menggunakan StackSets, dan menjawab pertanyaan umum tentang cara bekerja dengan dan memecahkan masalah StackSet pembuatan, pembaruan, dan penghapusan.

Topik

- [StackSets konsep](#)
- [Prasyarat untuk menggunakan AWS CloudFormation StackSets](#)
- [Memulai dengan StackSets menggunakan template sampel](#)
- [Buat AWS CloudFormation StackSets dengan izin yang dikelola sendiri](#)

- [Buat AWS CloudFormation StackSets dengan izin yang dikelola layanan](#)
- [Mengaktifkan atau menonaktifkan penerapan otomatis untuk in StackSets AWS Organizations](#)
- [Perbarui AWS CloudFormation StackSets](#)
- [Tambahkan tumpukan ke CloudFormation StackSets](#)
- [Ganti nilai parameter pada tumpukan di dalam CloudFormation StackSet](#)
- [Hapus tumpukan dari AWS CloudFormation StackSets](#)
- [Hapus AWS CloudFormation StackSets](#)
- [Mencegah StackSets penerapan yang gagal menggunakan gerbang akun target](#)
- [Pilih Mode Konkurensi untuk AWS CloudFormation StackSets](#)
- [Melakukan deteksi drift pada CloudFormation StackSets](#)
- [Impor tumpukan ke AWS CloudFormation StackSets](#)
- [Praktik terbaik untuk menggunakan AWS CloudFormation StackSets](#)
- [AWS CloudFormation StackSets contoh template](#)
- [Pemecahan masalah AWS CloudFormation StackSets](#)

StackSets konsep

Saat Anda menggunakan StackSets, Anda bekerja dengan StackSets, menumpuk instance, dan tumpukan.

Topik

- [Administrator dan akun target](#)
- [AWS CloudFormation StackSets](#)
- [Model izin untuk StackSets](#)
- [Instans tumpukan](#)
- [StackSet operasi](#)
- [StackSet opsi operasi](#)
- [Tanda](#)
- [StackSets kode status](#)
- [Kode status instance tumpukan](#)

Administrator dan akun target

Akun administrator adalah Akun AWS tempat Anda membuat StackSets. Untuk StackSets dengan izin yang dikelola layanan, akun administrator adalah akun manajemen organisasi atau akun administrator yang didelegasikan. Anda dapat mengelola StackSet dengan masuk ke akun AWS administrator yang membuat file StackSet.

Akun target adalah akun tempat Anda membuat, memperbarui, atau menghapus satu atau lebih tumpukan di akun Anda StackSet. Sebelum Anda dapat menggunakan StackSet untuk membuat tumpukan di akun target, siapkan hubungan kepercayaan antara administrator dan akun target.

AWS CloudFormation StackSets

A StackSet memungkinkan Anda membuat tumpukan di Akun AWS seluruh wilayah dengan menggunakan satu CloudFormation templat. CloudFormation Template A StackSet mendefinisikan semua sumber daya di setiap tumpukan. Saat Anda membuat StackSet, tentukan template yang akan digunakan, selain parameter dan kemampuan apa pun yang dibutuhkan template.

Setelah Anda menetapkan StackSet, Anda dapat membuat, memperbarui, atau menghapus tumpukan di akun target dan Wilayah AWS Anda menentukan. Saat Anda membuat, memperbarui, atau menghapus tumpukan, Anda juga dapat menentukan preferensi operasi. Misalnya, sertakan urutan Wilayah yang ingin Anda lakukan operasi, ambang toleransi kegagalan sebelum operasi tumpukan berhenti, dan jumlah akun yang melakukan operasi tumpukan secara bersamaan.

A StackSet adalah sumber daya regional. Jika Anda membuat StackSet dalam satu Wilayah AWS, Anda hanya dapat melihat atau mengubahnya saat melihat Wilayah itu.

Model izin untuk StackSets

Anda dapat membuat StackSets menggunakan izin yang dikelola sendiri atau izin yang dikelola layanan.

Dengan izin yang dikelola sendiri, Anda membuat peran IAM yang diperlukan StackSets untuk menyebarkan di seluruh akun dan Wilayah. Peran ini diperlukan untuk membangun hubungan tepercaya antara akun tempat Anda mengelola StackSet dari dan akun tempat Anda menerapkan instance tumpukan. Menggunakan model izin ini, StackSets dapat menyebarkan ke mana pun Akun AWS di mana Anda memiliki izin untuk membuat peran IAM.

Dengan izin yang dikelola layanan, Anda dapat men-deploy instans tumpukan ke akun yang dikelola oleh AWS Organizations. Dengan menggunakan model izin ini, Anda tidak perlu membuat peran

IAM yang diperlukan; StackSets membuat peran IAM atas nama Anda. Dengan model ini, Anda juga dapat mengaktifkan penerapan otomatis ke akun yang Anda tambahkan ke organisasi Anda di masa mendatang.

AWS Organizations terintegrasi dengan CloudFormation dan membantu Anda mengelola dan mengatur lingkungan secara terpusat saat Anda mengukur dan menumbuhkan sumber daya Anda. AWS

- Akun manajemen — akun yang Anda gunakan untuk membuat organisasi. Untuk informasi lebih lanjut, lihat [Terminologi dan konsep untuk AWS Organizations](#).
- Administrator yang didelegasikan — AWS layanan yang kompatibel dapat mendaftarkan akun AWS anggota di organisasi sebagai administrator untuk akun organisasi dalam layanan tersebut. Untuk informasi selengkapnya, lihat [AWS layanan yang dapat Anda gunakan AWS Organizations](#).

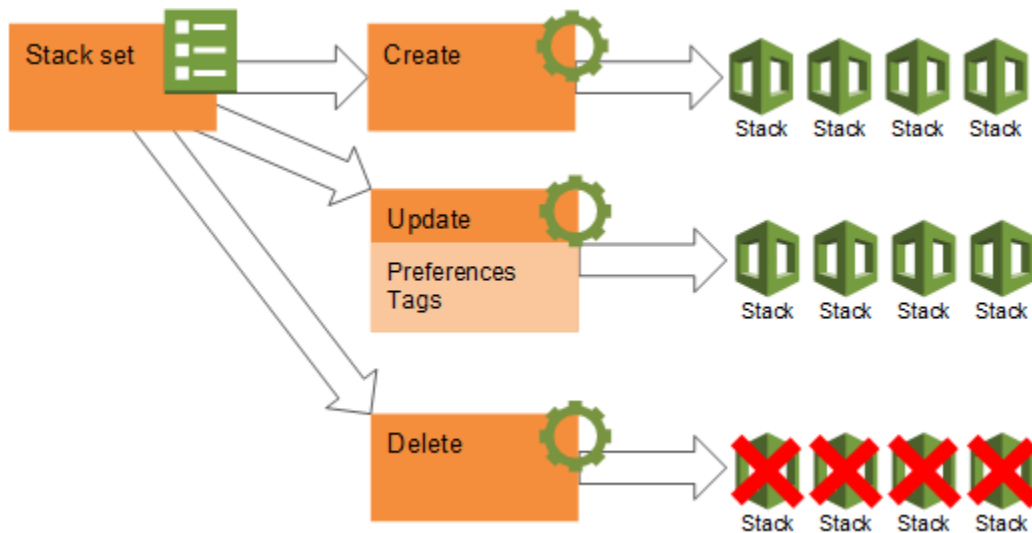
Untuk informasi selengkapnya tentang membuat dan mengelola StackSets dengan izin yang dikelola layanan, lihat topik berikut:

- [Aktifkan akses tepercaya untuk StackSets dengan AWS Organizations](#)
- [Daftarkan akun anggota administrator yang didelegasikan](#)
- [Buat AWS CloudFormation StackSets dengan izin yang dikelola layanan](#)

Instans tumpukan

Instans tumpukan adalah referensi ke tumpukan di akun target dalam Wilayah. Sebuah instance stack bisa ada tanpa tumpukan. Misalnya, jika tumpukan tidak dapat dibuat karena alasan tertentu, instance tumpukan menunjukkan alasan kegagalan pembuatan tumpukan. Sebuah instance stack mengasosiasikan dengan hanya satu StackSet.

Gambar berikut menunjukkan hubungan logis antara StackSets, operasi tumpukan, dan tumpukan. Saat Anda memperbarui StackSet, semua instance tumpukan terkait akan diperbarui di seluruh akun dan Wilayah.



StackSet operasi

Anda dapat melakukan operasi berikut pada StackSets.

Buat StackSet

Membuat yang baru StackSet termasuk menentukan CloudFormation templat yang ingin Anda gunakan untuk membuat tumpukan, menentukan akun target tempat Anda ingin membuat tumpukan, dan mengidentifikasi tempat Anda ingin menerapkan tumpukan Wilayah AWS di akun target Anda. A StackSet memastikan penerapan sumber daya tumpukan yang sama secara konsisten, dengan pengaturan yang sama, ke semua akun target yang ditentukan dalam Wilayah yang Anda pilih.

Perbarui StackSet

Ketika Anda memperbarui StackSet, Anda mendorong perubahan ke tumpukan di Anda StackSet. Anda dapat memperbarui StackSet dengan salah satu cara berikut. Pembaruan template Anda selalu memengaruhi semua tumpukan; Anda tidak dapat memperbarui template secara selektif untuk beberapa tumpukan di StackSet, tetapi tidak yang lain.

- Ubah setelan yang ada di templat atau tambahkan sumber daya baru, seperti memperbarui setelan parameter untuk layanan tertentu, atau menambahkan EC2 instans Amazon baru.
- Ganti templat dengan templat yang berbeda.
- Tambahkan tumpukan di akun target yang ada atau tambahan, di seluruh Wilayah yang ada atau tambahan.

Hapus tumpukan

Bila Anda menghapus tumpukan, Anda menghapus tumpukan dan semua sumber daya yang terkait dari akun target yang Anda tentukan, dalam Wilayah yang Anda tentukan. Anda dapat menghapus tumpukan dengan cara berikut.

- Hapus tumpukan dari beberapa akun target, sementara meninggalkan tumpukan lain di akun target lain berjalan.
- Hapus tumpukan dari beberapa Wilayah sambil meninggalkan tumpukan di Wilayah lain berjalan.
- Hapus tumpukan dari Anda StackSet, tetapi simpan agar terus berjalan secara independen StackSet dengan memilih opsi Retain Stacks. Anda kemudian dapat mengelola tumpukan yang dipertahankan di luar tempat Anda StackSet . CloudFormation
- Hapus semua tumpukan di Anda StackSet, sebagai persiapan untuk menghapus seluruh Anda. StackSet

Hapus StackSet

Anda dapat menghapus StackSet hanya ketika tidak ada instance tumpukan di dalamnya.

StackSet opsi operasi

Opsi yang dijelaskan di bagian ini membantu mengontrol waktu dan jumlah kegagalan yang diizinkan untuk melakukan StackSet operasi yang berhasil, dan mencegah Anda kehilangan sumber daya tumpukan.

Akun serentak maksimum

Pengaturan ini, tersedia dalam membuat, memperbarui, dan menghapus alur kerja, memungkinkan Anda menentukan jumlah maksimum atau persentase akun target di mana operasi dilakukan pada satu waktu. Angka atau persentase yang lebih rendah berarti bahwa operasi dilakukan di akun target yang lebih sedikit pada satu waktu. Operasi dilakukan di satu Wilayah pada satu waktu, dalam urutan yang ditentukan dalam kotak urutan Deployment. Misalnya, jika Anda menerapkan tumpukan ke 10 akun target dalam dua Wilayah, menyetel akun bersamaan Maksimum ke 50 dan Berdasarkan persentase menyebarkan tumpukan ke lima akun di Wilayah pertama, lalu lima akun kedua di Wilayah pertama, sebelum pindah ke Wilayah berikutnya dan memulai penerapan ke lima akun target pertama.

Bila Anda memilih Berdasarkan persentase, jika persentase yang ditentukan tidak mewakili seluruh jumlah akun yang Anda tentukan, CloudFormation bulatkan ke bawah. Misalnya, jika Anda menerapkan tumpukan ke 10 akun target, dan Anda menetapkan akun bersamaan Maksimum ke 25 dan persentase By, CloudFormation bulatkan ke bawah dari penerapan 2,5 tumpukan secara bersamaan (yang tidak mungkin) untuk menerapkan dua tumpukan secara bersamaan.

Perhatikan bahwa pengaturan ini memungkinkan Anda menentukan maksimum untuk operasi. Untuk deployment yang besar, dalam keadaan tertentu jumlah akun aktual yang ditindaklanjuti secara bersamaan mungkin lebih rendah karena throttling layanan.

Akun bersamaan maksimum dapat bergantung pada nilai toleransi Kegagalan tergantung pada Mode Konkurensi Anda. Jika Mode Konkurensi Anda diatur ke Toleransi Kegagalan Ketat maka akun bersamaan Maksimum dapat paling banyak satu lebih banyak daripada pengaturan toleransi Kegagalan.

Mode konkurensi

Pengaturan ini, tersedia dalam membuat, memperbarui, dan menghapus alur kerja, memungkinkan Anda memilih bagaimana tingkat konkurensi berperilaku selama operasi.

StackSet Untuk informasi selengkapnya, lihat [Pilih Mode Konkurensi untuk AWS CloudFormation StackSets](#).

Toleransi kegagalan

Pengaturan ini, tersedia dalam membuat, memperbarui, dan menghapus alur kerja, memungkinkan Anda menentukan jumlah maksimum atau persentase kegagalan operasi tumpukan yang dapat terjadi, per Wilayah, di luar yang CloudFormation menghentikan operasi secara otomatis. Angka atau persentase yang lebih rendah berarti operasi dilakukan pada tumpukan yang lebih sedikit, tetapi Anda dapat memulai pemecahan masalah operasi yang gagal lebih cepat. Misalnya, jika Anda memperbarui 10 tumpukan di 10 akun target dalam tiga wilayah, pengaturan Toleransi kegagalan ke 20 dan Menurut persentase berarti bahwa maksimum dua tumpukan pembaruan di wilayah dapat gagal melanjutkan operasi. Jika tumpukan ketiga di Wilayah yang sama gagal, CloudFormation hentikan operasi. Jika tumpukan tidak dapat diperbarui di Wilayah pertama, operasi pembaruan berlanjut di Wilayah tersebut, dan kemudian beralih ke Wilayah berikutnya. Jika dua tumpukan tidak dapat diperbarui di Wilayah kedua, toleransi kegagalan mencapai 20%; jika tumpukan ketiga di Wilayah gagal, CloudFormation hentikan operasi pembaruan, dan tidak melanjutkan ke Wilayah berikutnya.

Bila Anda memilih Berdasarkan persentase, jika persentase yang ditentukan tidak mewakili seluruh jumlah tumpukan Anda di setiap Wilayah, CloudFormation bulatkan ke bawah. Misalnya, jika Anda menerapkan tumpukan ke 10 akun target di tiga Wilayah, dan Anda menetapkan toleransi Kegagalan ke 25 dan By persentase, CloudFormation bulatkan dari toleransi kegagalan 2,5 tumpukan (yang tidak mungkin) menjadi toleransi kegagalan dua tumpukan per Wilayah.

Mempertahankan tumpukan

Pengaturan ini, tersedia dalam alur kerja delete stack, memungkinkan Anda menjaga tumpukan dan sumber dayanya tetap berjalan bahkan setelah menghapus tumpukan dari file. StackSet Ketika Anda menyimpan tumpukan, AWS CloudFormation meninggalkan tumpukan di akun individu dan Wilayah utuh. Tumpukan terpisah dari StackSet, tetapi menyimpan tumpukan dan sumber dayanya. Setelah operasi hapus tumpukan selesai, Anda mengelola tumpukan tertahan di CloudFormation, di akun target (bukan akun administrator) yang membuat tumpukan.

Konkurensi wilayah

Setelan ini, tersedia dalam membuat, memperbarui, dan menghapus alur kerja, memungkinkan Anda memilih cara StackSets menerapkan ke Wilayah.

Sequential — Menyebarkan StackSets operasi ke dalam satu Wilayah pada satu waktu seperti yang ditentukan oleh kotak urutan Penyebaran Wilayah selama kegagalan penerapan Wilayah tidak melebihi toleransi kegagalan yang ditentukan. Deployment berurutan adalah pilihan default.

Paralel - Menerapkan StackSets operasi ke semua Wilayah tertentu secara bersamaan selama kegagalan penerapan Wilayah tidak melebihi toleransi kegagalan yang ditentukan.

Tanda

Anda dapat menambahkan tag selama operasi StackSet pembuatan dan pembaruan dengan menentukan pasangan kunci dan nilai. Tag berguna untuk menyortir dan memfilter StackSet sumber daya untuk penagihan dan alokasi biaya. Untuk informasi selengkapnya tentang cara menggunakan tag AWS, lihat [Mengatur dan melacak AWS biaya menggunakan tag alokasi](#) biaya di Panduan AWS Manajemen Penagihan dan Biaya Pengguna. Setelah Anda menentukan pasangan nilai kunci, pilih + untuk menyimpan tag. Anda dapat menghapus tag yang sudah tidak Anda gunakan dengan memilih X merah di sebelah kanan tag.

Tag yang Anda StackSets terapkan untuk diterapkan ke semua tumpukan, dan ke sumber daya yang dibuat oleh tumpukan Anda. Anda juga dapat menambahkan tag pada level stack-only di CloudFormation, tetapi tag tersebut mungkin tidak muncul. StackSets

Meskipun StackSets tidak menambahkan tag yang ditentukan sistem, Anda tidak boleh memulai nama kunci dari tag apa pun dengan string. `aws :`

StackSets kode status

AWS CloudFormation StackSets menghasilkan kode status untuk StackSet operasi.

Tabel berikut menjelaskan kode status untuk StackSet operasi.

RUNNING

Operasi saat ini sedang berlangsung.

SUCCEEDED

Operasi selesai tanpa melebihi toleransi kegagalan untuk operasi.

FAILED

Jumlah tumpukan yang tidak dapat diselesaikan operasi melebihi toleransi kegagalan yang ditentukan pengguna. Nilai toleransi kegagalan yang telah Anda tetapkan untuk operasi berlaku untuk setiap Wilayah selama pembuatan tumpukan dan operasi pembaruan. Jika jumlah tumpukan yang gagal dalam suatu Wilayah melebihi toleransi kegagalan, status operasi di Wilayah berubah menjadi `FAILED`. Status operasi secara keseluruhan juga diatur ke `FAILED`, dan CloudFormation membatalkan operasi di Wilayah yang tersisa.

QUEUED

[Service-managed permissions] Untuk penerapan otomatis yang memerlukan urutan operasi, operasi masuk ke antrian untuk dilakukan. Misalnya:

- Memindahkan akun dari satu unit organisasi (OU) `OU1`, ke yang lain `OU2`, memicu penyebaran otomatis. StackSets menjalankan operasi penghapusan untuk menghapus instance tumpukan dari `OU1` akun target di Wilayah target dan mengantri operasi buat untuk menambahkan instance tumpukan ke `OU2` akun target di Wilayah target.
- Menambahkan akun `AccountA` ke OU memicu penyebaran otomatis. StackSets menjalankan operasi `create` untuk menambahkan instance stack ke `AccountA` Region target. Jika Anda menambahkan akun lain `AccountB` ke OU saat operasi buat ini berjalan, StackSets antrian operasi buat kedua. Ketika operasi `create` pertama selesai, StackSets jalankan operasi `create` kedua untuk menambahkan instance stack ke `AccountB` Region target.

STOPPING

Operasi sedang dalam proses berhenti, atas permintaan pengguna.

STOPPED

Operasi telah berhenti, atas permintaan pengguna.

Kode status instance tumpukan

AWS CloudFormation StackSets menghasilkan kode status untuk instance tumpukan.

Tabel berikut menjelaskan kode status untuk instance stack di dalamnya StackSets.

CURRENT

Tumpukan up to date dengan StackSet.

OUTDATED

Tumpukan tidak up to date dengan salah satu StackSet alasan berikut.

- Suatu [CreateStackSet](#) atau [UpdateStackSet](#) operasi pada tumpukan terkait gagal.
- Tumpukan adalah bagian dari [UpdateStackSet](#) operasi [CreateStackSet](#) atau yang gagal, atau berhenti sebelum membuat atau memperbarui tumpukan.

INOPERABLE

Operasi [DeleteStackInstances](#) telah gagal dan meninggalkan tumpukan dalam keadaan tidak stabil. Tumpukan di status ini dikecualikan dari [UpdateStackSet](#) operasi lebih lanjut. Anda mungkin harus melakukan [DeleteStackInstances](#) operasi, dengan `RetainStacks` diatur ke `true`, untuk menghapus instans tumpukan, dan kemudian menghapus tumpukan secara manual.

CANCELLED

Operasi di akun yang ditentukan dan Wilayah telah dibatalkan. Ini terjadi karena pengguna telah menghentikan StackSet operasi, atau karena StackSet operasi melebihi toleransi kegagalan.

FAILED

Operasi di akun yang ditentukan dan Wilayah gagal. Jika StackSet operasi gagal dalam akun yang cukup dalam suatu Wilayah, toleransi kegagalan untuk StackSet operasi secara keseluruhan mungkin terlampaui.

FAILED_IMPORT

Impor instance tumpukan di akun dan Wilayah yang ditentukan gagal dan meninggalkan tumpukan dalam keadaan tidak stabil. Setelah masalah yang menyebabkan kegagalan diperbaiki,

operasi impor dapat dicoba lagi. Jika StackSet operasi yang cukup gagal dalam akun yang cukup dalam suatu Wilayah, toleransi kegagalan untuk StackSet operasi secara keseluruhan mungkin terlampaui.

PENDING

Operasi di akun dan Wilayah tertentu belum dimulai.

RUNNING

Operasi di akun dan Wilayah tertentu saat ini sedang berlangsung.

SKIPPED_SUSPENDED_ACCOUNT

Operasi di akun dan Wilayah yang ditentukan telah dilewati karena akun ditangguhkan pada saat operasi.

SUCCEEDED

Operasi di akun dan Region tertentu selesai dengan sukses.

Prasyarat untuk menggunakan AWS CloudFormation StackSets

StackSets memperluas fungsionalitas tumpukan, sehingga Anda dapat membuat, memperbarui, atau menghapus tumpukan di beberapa akun dan Wilayah dengan satu operasi.

Karena StackSets melakukan operasi tumpukan di beberapa akun, sebelum Anda dapat membuat yang pertama, StackSet Anda memerlukan izin yang diperlukan yang ditentukan dalam akun Anda Akun AWS.

Anda dapat mengelola StackSets menggunakan izin yang dikelola sendiri atau dikelola layanan.

- Untuk dikelola sendiri StackSets, Anda harus membuat dan mengelola peran IAM di setiap akun target dan. Wilayah AWS Untuk informasi selengkapnya, lihat [Memberikan izin yang dikelola sendiri](#).
- Untuk pengelolaan layanan StackSets, Anda tidak perlu membuat dan mengelola peran IAM secara manual di setiap akun; AWS menangani pembuatan peran dan izin untuk Anda. Untuk informasi selengkapnya, lihat [Aktifkan akses tepercaya](#).

 Note

Mengaktifkan akses tepercaya dengan AWS Organizations for saat ini AWS CloudFormation StackSets tidak didukung di Wilayah Beijing dan Ningxia Tiongkok.

Topik

- [Bersiaplah untuk melakukan StackSet operasi Wilayah AWS yang dinonaktifkan secara default](#)
- [Memberikan izin yang dikelola sendiri](#)
- [Aktifkan akses tepercaya untuk StackSets dengan AWS Organizations](#)

Bersiaplah untuk melakukan StackSet operasi Wilayah AWS yang dinonaktifkan secara default

Wilayah AWS diperkenalkan setelah 20 Maret 2019, seperti Asia Pasifik (Hong Kong), dinonaktifkan secara default. Anda harus mengaktifkan Wilayah ini untuk akun Anda sebelum dapat menggunakannya. Untuk informasi selengkapnya, lihat [Mengaktifkan atau menonaktifkan Wilayah AWS di akun Anda](#) di Panduan AWS Account Management Referensi.

Untuk membuat StackSet dari akun administrator (jika menggunakan izin yang dikelola sendiri) atau akun manajemen organisasi (jika menggunakan izin yang dikelola layanan) di Wilayah yang dinonaktifkan secara default, Anda harus mengaktifkan Wilayah tersebut terlebih dahulu untuk akun administrator atau manajemen. StackSet

AWS CloudFormation Untuk berhasil membuat atau memperbarui instance tumpukan:

- Akun target harus berada di Wilayah yang saat ini diaktifkan untuk akun target tersebut.
- Akun administrator atau akun manajemen organisasi harus mengaktifkan Wilayah yang sama dengan akun target. StackSet

 Important

Ketahui bahwa selama StackSet operasi, administrator dan akun target bertukar metadata mengenai akun itu sendiri, selain StackSet dan StackSet contoh yang terlibat.

Selain itu, jika Anda menonaktifkan Wilayah yang berisi akun di mana StackSet instans berada, Anda bertanggung jawab untuk menghapus instans atau sumber daya tersebut, jika

diinginkan. Selain itu, ketahuilah bahwa metadata mengenai akun target di Wilayah yang dinonaktifkan akan disimpan di akun administrator.

Memberikan izin yang dikelola sendiri

Topik ini memberikan petunjuk tentang cara membuat peran layanan IAM yang diperlukan StackSets untuk menyebarkan di seluruh akun dan Wilayah AWS dengan izin yang dikelola sendiri. Peran ini diperlukan untuk membangun hubungan tepercaya antara akun tempat Anda mengelola StackSet dari dan akun tempat Anda menerapkan instance tumpukan. Menggunakan model izin ini, StackSets dapat menyebarkan ke mana pun Akun AWS di mana Anda memiliki izin untuk membuat peran IAM.

Untuk menggunakan izin yang dikelola layanan, lihat sebagai gantinya. [Aktifkan akses tepercaya](#)

Topik

- [Ikhtisar izin yang dikelola sendiri](#)
- [Berikan semua pengguna izin akun administrator untuk mengelola tumpukan di semua akun target](#)
- [Siapkan opsi izin lanjutan untuk operasi StackSet](#)
- [Siapkan kunci global untuk mengurangi masalah wakil yang membingungkan](#)

Ikhtisar izin yang dikelola sendiri

Sebelum membuat izin StackSet yang dikelola sendiri, Anda harus telah membuat peran layanan IAM di setiap akun.

Langkah dasarnya adalah:

1. Tentukan yang Akun AWS merupakan akun administrator.

StackSets dibuat di akun administrator ini. Akun target adalah akun tempat Anda membuat tumpukan individual milik a StackSet.

2. Tentukan bagaimana Anda ingin menyusun izin untuk. StackSet

Konfigurasi izin yang paling sederhana (dan paling permisif) adalah di mana Anda memberi semua pengguna dan grup di akun administrator kemampuan untuk membuat dan memperbarui semua yang StackSets dikelola melalui akun itu. Jika Anda memerlukan kendali yang lebih halus, Anda dapat mengatur izin untuk menentukan:

- Pengguna dan grup mana yang dapat melakukan StackSet operasi di mana akun target.

- Sumber daya mana yang dapat disertakan oleh pengguna dan grup di dalamnya StackSets.
 - StackSet Operasi mana yang dapat dilakukan pengguna dan grup tertentu.
3. Buat peran layanan IAM yang diperlukan di administrator dan akun target Anda untuk menentukan izin yang Anda inginkan.

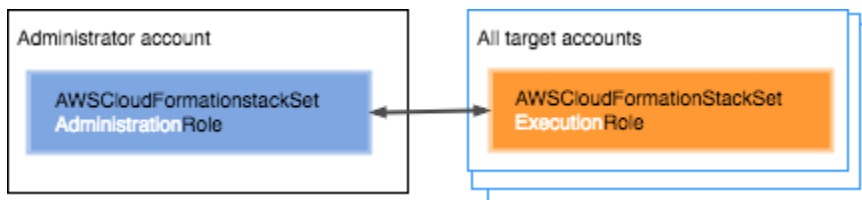
Secara khusus, dua peran yang diperlukan adalah:

- `AWSCloudFormationStackSetAdministrationRole`— Peran ini diterapkan ke akun administrator.
- `AWSCloudFormationStackSetExecutionRole`— Peran ini diterapkan ke semua akun tempat Anda membuat instance tumpukan.

Berikan semua pengguna izin akun administrator untuk mengelola tumpukan di semua akun target

Bagian ini menunjukkan cara mengatur izin agar semua pengguna dan grup akun administrator dapat melakukan StackSet operasi di semua akun target. Ini memandu Anda melalui pembuatan peran layanan IAM yang diperlukan di administrator dan akun target Anda. Siapa pun di akun administrator kemudian dapat membuat, memperbarui, atau menghapus tumpukan apa pun di salah satu akun target.

Dengan menyusun izin dengan cara ini, pengguna tidak melewati peran administrasi saat membuat atau memperbarui file. StackSet



Administrator account

Di akun administrator, buat peran IAM bernama `AWSCloudFormationStackSetAdministrationRole`.

Anda dapat melakukan ini dengan membuat tumpukan dari CloudFormation template yang tersedia <https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/AWSCloudFormationStackSetAdministrationRole.dari.yml>.

Example Contoh kebijakan izin

Peran administrasi yang dibuat oleh template sebelumnya mencakup kebijakan izin berikut.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "sts:AssumeRole"
      ],
      "Resource": [
        "arn:aws:iam::*:role/AWSCloudFormationStackSetExecutionRole"
      ],
      "Effect": "Allow"
    }
  ]
}
```

Example Contoh kebijakan kepercayaan 1

Templat sebelumnya juga menyertakan kebijakan kepercayaan berikut yang memberikan izin layanan untuk menggunakan peran administrasi dan izin yang dilampirkan pada peran tersebut.

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

Example Contoh kebijakan kepercayaan 2

Untuk menerapkan instance tumpukan ke akun target yang berada di Wilayah yang dinonaktifkan secara default, Anda juga harus menyertakan prinsip layanan regional untuk Wilayah tersebut. Setiap Wilayah yang dinonaktifkan secara default akan memiliki prinsip layanan regionalnya sendiri.

Contoh kebijakan kepercayaan berikut memberikan izin layanan untuk menggunakan peran administrasi di Wilayah Asia Pasifik (Hong Kong-east-1), Wilayah yang dinonaktifkan secara default.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "cloudformation.amazonaws.com",
          "cloudformation.ap-east-1.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Untuk informasi selengkapnya, lihat [Bersiaplah untuk melakukan StackSet operasi Wilayah AWS yang dinonaktifkan secara default](#). Untuk daftar kode Wilayah, lihat [Titik akhir Regional](#) di Referensi Umum AWS Panduan.

Target accounts

Di setiap akun target, buat peran layanan bernama `AWSCloudFormationStackSetExecutionRole` yang mempercayai akun administrator. Peran ini harus memiliki nama yang tepat. Anda dapat melakukan ini dengan membuat tumpukan dari CloudFormation template yang tersedia <https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/AWSCloudFormationStackSetExecutionRole.dari.yl>. Saat Anda menggunakan templat ini, Anda diminta untuk memberikan ID akun administrator yang dengannya akun target Anda harus memiliki hubungan kepercayaan.

Important

Ketahui bahwa templat ini memberikan akses administrator. Setelah Anda menggunakan template untuk membuat peran eksekusi akun target, Anda harus mencakup izin dalam pernyataan kebijakan ke jenis sumber daya yang Anda buat dengan menggunakan StackSets.

Peran layanan akun target memerlukan izin untuk melakukan operasi apa pun yang ditentukan dalam CloudFormation templat Anda. Misalnya, jika templat Anda membuat bucket S3, Anda akan memerlukan izin untuk membuat objek baru untuk S3. Akun target Anda selalu memerlukan izin CloudFormation penuh, yang mencakup izin untuk membuat, memperbarui, menghapus, dan menjelaskan tumpukan.

Example Contoh kebijakan izin 1

Peran yang dibuat oleh templat ini mengaktifkan kebijakan berikut di akun target Anda.

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

Example Contoh kebijakan izin 2

Contoh berikut menunjukkan pernyataan kebijakan dengan izin minimum StackSets untuk bekerja. Untuk membuat tumpukan di akun target yang menggunakan sumber daya dari layanan selain CloudFormation, Anda harus menambahkan tindakan layanan dan sumber daya tersebut ke pernyataan `AWSCloudFormationStackSetExecutionRolekebijakan` untuk setiap akun target.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudformation:*"
      ],
      "Resource": "*"
    }
  ]
}
```

Example Contoh kebijakan kepercayaan

Hubungan kepercayaan berikut dibuat oleh templat. ID akun administrator ditampilkan sebagai *admin_account_id*.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::admin_account_id:root"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Anda dapat mengonfigurasi hubungan kepercayaan peran eksekusi akun target yang ada untuk mempercayai peran tertentu di akun administrator. Jika Anda menghapus peran di akun administrator, dan membuat yang baru untuk menggantinya, Anda harus mengonfigurasi hubungan kepercayaan akun target Anda dengan peran akun administrator baru, yang diwakili oleh *admin_account_id* dalam contoh sebelumnya.

Siapkan opsi izin lanjutan untuk operasi StackSet

Jika Anda memerlukan kontrol yang lebih halus atas pengguna dan grup StackSets yang dibuat melalui satu akun administrator, Anda dapat menggunakan peran IAM untuk menentukan:

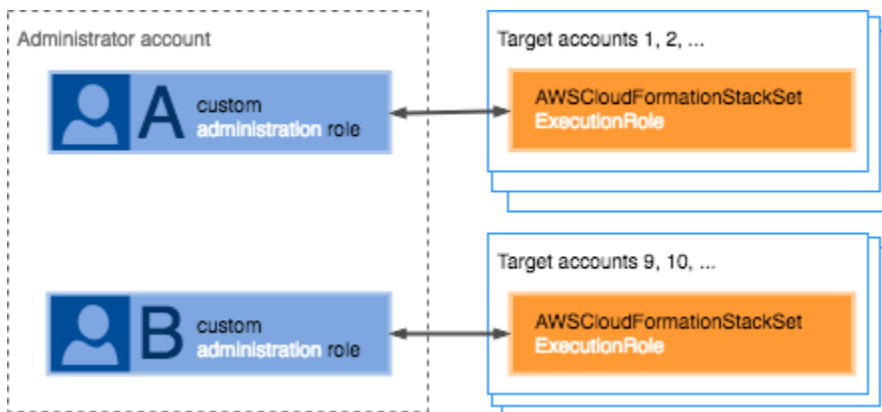
- Pengguna dan grup mana yang dapat melakukan StackSet operasi di mana akun target.
- Sumber daya mana yang dapat disertakan oleh pengguna dan grup di dalamnya StackSets.
- StackSet Operasi mana yang dapat dilakukan pengguna dan grup tertentu.

Kontrol pengguna mana yang dapat melakukan StackSet operasi di akun target tertentu

Gunakan peran administrasi yang disesuaikan untuk mengontrol pengguna dan grup mana yang dapat melakukan StackSet operasi di akun target mana. Anda mungkin ingin mengontrol pengguna akun administrator mana yang dapat melakukan StackSet operasi di akun target mana. Untuk melakukan ini, Anda membuat hubungan kepercayaan antara

setiap akun target dan peran administrasi khusus khusus, daripada membuat peran `AWSCloudFormationStackSetAdministrationRole` layanan di akun administrator itu sendiri. Anda kemudian mengaktifkan pengguna dan grup tertentu untuk menggunakan peran administrasi yang disesuaikan saat melakukan StackSet operasi di akun target tertentu.

Misalnya, Anda dapat membuat Peran A dan Peran B dalam akun administrator. Anda dapat memberikan izin Peran A untuk mengakses akun target 1 melalui akun 8. Anda dapat memberikan izin Peran B untuk mengakses akun target 9 melalui akun 16.



Menyiapkan izin yang diperlukan melibatkan penentuan peran administrasi yang disesuaikan, membuat peran layanan untuk akun target, dan memberikan izin kepada pengguna untuk lulus peran administrasi yang disesuaikan saat melakukan operasi. StackSet

Secara umum, inilah cara kerjanya setelah Anda memiliki izin yang diperlukan: Saat membuat StackSet, pengguna harus menentukan peran administrasi yang disesuaikan. Pengguna harus memiliki izin untuk meneruskan peran ke CloudFormation. Selain itu, peran administrasi yang disesuaikan harus memiliki hubungan kepercayaan dengan akun target yang ditentukan untuk akun StackSet. CloudFormation menciptakan StackSet dan mengaitkan peran administrasi yang disesuaikan dengannya. Saat memperbarui StackSet, pengguna harus secara eksplisit menentukan peran administrasi yang disesuaikan, meskipun itu adalah peran administrasi khusus yang sama yang digunakan dengan ini StackSet sebelumnya. CloudFormation menggunakan peran itu untuk memperbarui tumpukan, tunduk pada persyaratan di atas.

Administrator account

Example Contoh kebijakan izin

Untuk masing-masing StackSet, buat peran administrasi yang disesuaikan dengan izin untuk mengambil peran eksekusi akun target.

Nama peran eksekusi akun target harus sama di setiap akun target. Jika nama peran `AWSCloudFormationStackSetExecutionRole`, `StackSets` gunakan secara otomatis saat membuat file `StackSet`. Jika Anda menentukan nama peran kustom, pengguna harus memberikan nama peran eksekusi saat membuat `StackSet`.

Buat [peran layanan IAM](#) dengan nama kustom dan kebijakan izin berikut. Dalam contoh di bawah ini, `custom_execution_role` mengacu pada peran eksekusi di akun target.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "sts:AssumeRole"
      ],
      "Resource": [
        "arn:aws:iam::target_account_id:role/custom_execution_role"
      ],
      "Effect": "Allow"
    }
  ]
}
```

Untuk menentukan beberapa akun dalam satu pernyataan, pisahkan dengan koma.

```
"Resource": [
  "arn:aws:iam::target_account_id_1:role/custom_execution_role",
  "arn:aws:iam::target_account_id_2:role/custom_execution_role"
]
```

Anda dapat menentukan semua akun target dengan menggunakan wildcard (*), bukan ID akun.

```
"Resource": [
  "arn:aws:iam::*:role/custom_execution_role"
]
```

Example Contoh kebijakan kepercayaan 1

Anda harus memberikan kebijakan kepercayaan untuk peran layanan untuk menentukan prinsip IAM mana yang dapat mengambil peran tersebut.

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

Example Contoh kebijakan kepercayaan 2

Untuk menerapkan instance tumpukan ke akun target yang berada di Wilayah yang dinonaktifkan secara default, Anda juga harus menyertakan prinsip layanan regional untuk Wilayah tersebut. Setiap Wilayah yang dinonaktifkan secara default akan memiliki prinsip layanan regionalnya sendiri.

Contoh kebijakan kepercayaan berikut memberikan izin layanan untuk menggunakan peran administrasi di Wilayah Asia Pasifik (Hong Kongap-east-1), Wilayah yang dinonaktifkan secara default.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "cloudformation.amazonaws.com",
          "cloudformation.ap-east-1.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Untuk informasi selengkapnya, lihat [Bersiaplah untuk melakukan StackSet operasi Wilayah AWS yang dinonaktifkan secara default](#). Untuk daftar kode Wilayah, lihat [Titik akhir Regional](#) di Panduan Referensi AWS Umum.

Example Contoh kebijakan peran lulus

Anda juga memerlukan kebijakan izin IAM untuk pengguna IAM Anda yang memungkinkan pengguna untuk lulus peran administrasi yang disesuaikan saat melakukan operasi. StackSet Untuk informasi selengkapnya, lihat [Memberikan izin pengguna untuk meneruskan peran ke layanan AWS](#).

Dalam contoh di bawah ini, *customized_admin_role* mengacu pada peran administrasi yang harus dilewati pengguna.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iam:GetRole",
        "iam:PassRole"
      ],
      "Resource": "arn:aws:iam::*:role/customized_admin_role"
    }
  ]
}
```

Target accounts

Di setiap akun target, buat peran layanan yang mempercayai peran administrasi khusus yang ingin Anda gunakan dengan akun ini.

Peran akun target memerlukan izin untuk melakukan operasi apa pun yang ditentukan dalam CloudFormation templat Anda. Misalnya, jika templat Anda membuat bucket S3, Anda akan memerlukan izin untuk membuat objek baru di S3. Akun target Anda selalu membutuhkan CloudFormation izin penuh, yang mencakup izin untuk membuat, memperbarui, menghapus, dan menjelaskan tumpukan.

Nama peran akun target harus sama di setiap akun target. Jika nama peran `AWSCloudFormationStackSetExecutionRole`, StackSets gunakan secara otomatis saat membuat

file StackSet. Jika Anda menentukan nama peran kustom, pengguna harus memberikan nama peran eksekusi saat membuat StackSet.

Example Contoh kebijakan izin

Contoh berikut menunjukkan pernyataan kebijakan dengan izin minimum StackSets untuk bekerja. Untuk membuat tumpukan di akun target yang menggunakan sumber daya dari layanan selain CloudFormation, Anda harus menambahkan tindakan layanan dan sumber daya tersebut ke kebijakan izin.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudformation:*"
      ],
      "Resource": "*"
    }
  ]
}
```

Example Contoh kebijakan kepercayaan

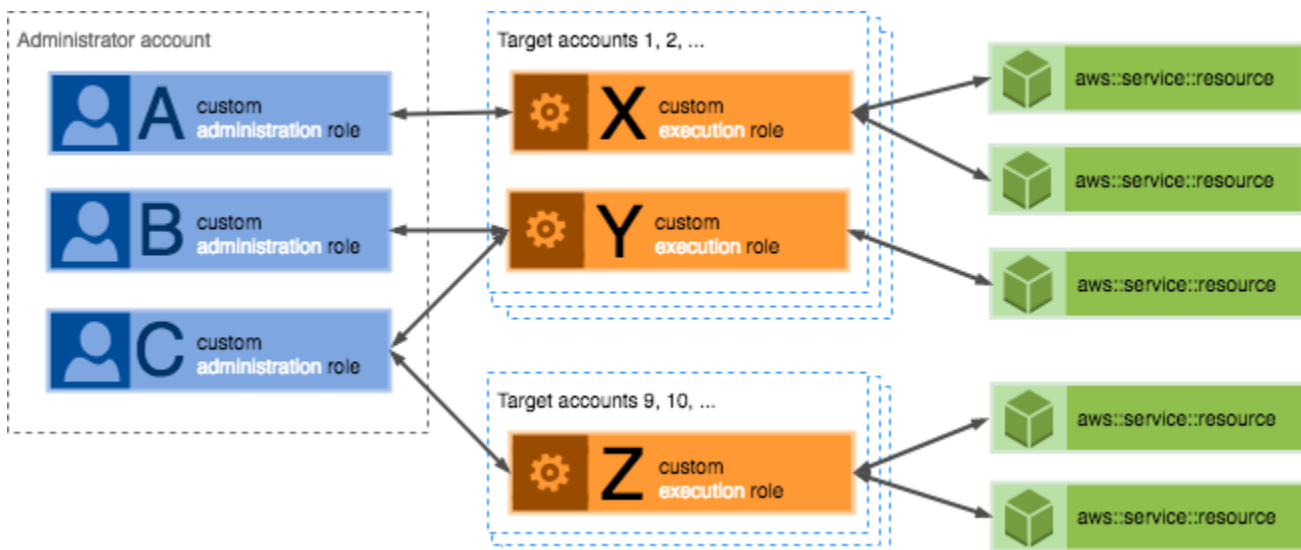
Anda harus memberikan kebijakan kepercayaan berikut ketika Anda membuat peran untuk menentukan hubungan kepercayaan.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::admin_account_id:role/customized_admin_role"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```


Kontrol sumber daya yang dapat disertakan pengguna secara spesifik StackSets

Gunakan peran eksekusi yang disesuaikan untuk mengontrol sumber daya tumpukan yang dapat disertakan oleh pengguna dan grup di dalamnya StackSets. Misalnya, Anda mungkin ingin menyiapkan grup yang hanya dapat menyertakan sumber daya terkait Amazon S3 dalam pembuatannya, sementara tim lain hanya dapat menyertakan sumber daya DynamoDB. StackSets Untuk melakukan ini, Anda membuat hubungan kepercayaan antara peran administrasi yang disesuaikan untuk setiap grup dan peran eksekusi yang disesuaikan untuk setiap kumpulan sumber daya. Peran eksekusi yang disesuaikan menentukan sumber daya tumpukan mana yang dapat disertakan. StackSets Peran administrasi yang disesuaikan berada di akun administrator, sedangkan peran eksekusi yang disesuaikan berada di setiap akun target yang ingin Anda buat StackSets menggunakan sumber daya yang ditentukan. Anda kemudian mengaktifkan pengguna dan grup tertentu untuk menggunakan peran administrasi yang disesuaikan saat melakukan StackSets operasi.

Misalnya Anda dapat membuat peran administrasi yang disesuaikan A, B, dan C di akun administrator. Pengguna dan grup dengan izin untuk menggunakan Peran A dapat membuat StackSets yang berisi sumber daya tumpukan yang secara khusus tercantum dalam peran eksekusi yang disesuaikan X, tetapi tidak dalam peran Y atau Z, atau sumber daya yang tidak termasuk dalam peran eksekusi apa pun.



Saat memperbarui StackSet, pengguna harus secara eksplisit menentukan peran administrasi yang disesuaikan, meskipun itu adalah peran administrasi khusus yang sama yang digunakan dengan ini StackSet sebelumnya. CloudFormation melakukan pembaruan menggunakan peran administrasi khusus yang ditentukan, selama pengguna memiliki izin untuk melakukan operasi pada itu StackSet.

Demikian pula, pengguna juga dapat menentukan peran eksekusi yang disesuaikan. Jika mereka menentukan peran eksekusi yang disesuaikan, CloudFormation gunakan peran itu untuk memperbarui tumpukan, tunduk pada persyaratan di atas. Jika pengguna tidak menentukan peran eksekusi yang disesuaikan, CloudFormation lakukan pembaruan menggunakan peran eksekusi khusus yang sebelumnya dikaitkan dengan StackSet, selama pengguna memiliki izin untuk melakukan operasi pada itu StackSet.

Administrator account

Buat peran administrasi yang disesuaikan di akun administrator Anda, seperti yang dijelaskan dalam [Kontrol pengguna mana yang dapat melakukan StackSet operasi di akun target tertentu](#). Sertakan hubungan kepercayaan antara peran administrasi yang disesuaikan dan peran eksekusi khusus yang ingin Anda gunakan.

Example Contoh kebijakan izin

Contoh berikut adalah kebijakan izin untuk kedua yang `AWSCloudFormationStackSetExecutionRole` ditentukan untuk akun target, selain peran eksekusi yang disesuaikan.

```
{
  "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

Di akun target tempat Anda ingin membuat StackSets, buat peran eksekusi khusus yang memberikan izin ke layanan dan sumber daya yang Anda inginkan agar pengguna dan grup dapat disertakan dalam akun. StackSets

Example Contoh kebijakan izin

Contoh berikut memberikan izin minimum untuk StackSets, bersama dengan izin untuk membuat tabel Amazon DynamoDB.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudformation:*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "dynamoDb:createTable"
      ],
      "Resource": "*"
    }
  ]
}
```

Example Contoh kebijakan kepercayaan

Anda harus memberikan kebijakan kepercayaan berikut ketika Anda membuat peran untuk menentukan hubungan kepercayaan.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::admin_account_id:role/customized_admin_role"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Menyiapkan izin untuk operasi tertentu StackSet

Selain itu, Anda dapat mengatur izin yang pengguna dan grup dapat melakukan StackSet operasi tertentu, seperti membuat, memperbarui, atau menghapus StackSets atau menumpuk instance. Untuk informasi selengkapnya, lihat [Kunci tindakan, sumber daya, dan kondisi CloudFormation](#) di Referensi Otorisasi Layanan.

Siapkan kunci global untuk mengurangi masalah wakil yang membingungkan

Masalah "confused deputy" adalah masalah keamanan saat entitas yang tidak memiliki izin untuk melakukan suatu tindakan dapat memaksa entitas yang memiliki hak akses lebih tinggi untuk melakukan tindakan tersebut. Pada tahun AWS, peniruan lintas layanan dapat mengakibatkan masalah wakil yang membingungkan. Peniruan identitas lintas layanan dapat terjadi ketika satu layanan (layanan yang dipanggil) memanggil layanan lain (layanan yang dipanggil). Layanan panggilan dapat dimanipulasi untuk menggunakan izinnya untuk bertindak atas sumber daya pelanggan lain dengan cara yang seharusnya tidak memiliki izin untuk mengakses. Untuk mencegah hal ini, AWS menyediakan alat yang membantu Anda melindungi data Anda untuk semua layanan dengan prinsip layanan yang telah diberikan akses ke sumber daya di akun Anda.

Sebaiknya gunakan kunci konteks kondisi [aws:SourceAccount](#) global [aws:SourceArn](#) dan dalam kebijakan sumber daya untuk membatasi izin yang AWS CloudFormation StackSets memberikan layanan lain ke sumber daya. Jika Anda menggunakan kedua kunci konteks kondisi global, `aws:SourceAccount` nilai dan akun dalam `aws:SourceArn` nilai harus menggunakan ID akun yang sama saat digunakan dalam pernyataan kebijakan yang sama.

Cara paling efektif untuk melindungi dari masalah "confused deputy" adalah dengan menggunakan kunci konteks kondisi global `aws:SourceArn` dengan ARN lengkap sumber daya. Jika Anda tidak mengetahui ARN lengkap sumber daya atau jika Anda menentukan beberapa sumber daya, gunakan kunci kondisi konteks `aws:SourceArn` global dengan wildcard (*) untuk bagian ARN yang tidak diketahui. Misalnya, `arn:aws:cloudformation::123456789012:*`. Bila memungkinkan, gunakan `aws:SourceArn`, karena lebih spesifik. Gunakan `aws:SourceAccount` hanya jika Anda tidak dapat menentukan pola ARN atau ARN yang benar.

Saat StackSets mengasumsikan peran Administrasi di akun administrator Anda, StackSets isi ID akun administrator Anda dan Nama Sumber Daya StackSets Amazon (ARN). Oleh karena itu, Anda dapat menentukan kondisi untuk [kunci global](#) `aws:SourceAccount` dan `aws:SourceArn` dalam hubungan kepercayaan untuk mencegah [masalah wakil yang membingungkan](#). Contoh berikut menunjukkan bagaimana Anda dapat menggunakan kunci konteks kondisi `aws:SourceAccount` global `aws:SourceArn` dan StackSets untuk mencegah masalah wakil yang membingungkan.

Administrator account

Example Kunci global untuk **aws:SourceAccount** dan **aws:SourceArn**

Saat menggunakan StackSets, tentukan kunci global `aws:SourceAccount` dan `aws:SourceArn` dalam kebijakan `AWSCloudFormationStackSetAdministrationRolekepercayaan` Anda untuk mencegah masalah wakil yang membingungkan.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "cloudformation.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "111122223333"
        },
        "StringLike": {
          "aws:SourceArn": "arn:aws:cloudformation:*:111122223333:stackset/*"
        }
      }
    }
  ]
}
```

Example StackSets ARNs

Tentukan terkait Anda StackSets ARNs untuk kontrol yang lebih baik.

```
{
  "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",
    ]
  }
}
```

Aktifkan akses tepercaya untuk StackSets dengan AWS Organizations

Topik ini memberikan petunjuk tentang cara mengaktifkan akses tepercaya AWS Organizations, yang diperlukan oleh StackSets untuk menyebarkan di seluruh akun dan Wilayah AWS menggunakan izin yang dikelola layanan. Untuk menggunakan izin yang dikelola sendiri, lihat [Memberikan izin yang dikelola sendiri](#) sebagai gantinya.

Sebelum Anda membuat izin yang dikelola layanan StackSet dengan layanan, Anda harus terlebih dahulu menyelesaikan tugas-tugas berikut:

- [Aktifkan semua fitur](#) di AWS Organizations. Dengan hanya fitur penagihan gabungan yang diaktifkan, Anda tidak dapat membuat izin StackSet dengan layanan yang dikelola.
- Aktifkan akses tepercaya dengan AWS Organizations. Tindakan ini memungkinkan CloudFormation untuk membuat peran terkait layanan di akun manajemen. Setelah akses tepercaya diaktifkan, saat Anda membuat StackSet dengan izin yang dikelola layanan, CloudFormation buat peran terkait layanan yang diperlukan dan peran layanan yang disebutkan `stacksets-exec-*` di akun target (anggota).

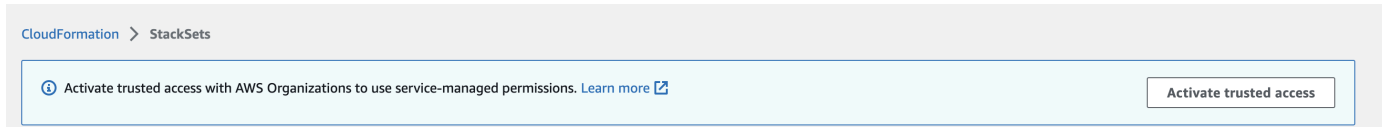
Dengan akses tepercaya diaktifkan, akun manajemen dan akun administrator yang didelegasikan dapat membuat dan mengelola layanan yang dikelola StackSets untuk organisasi mereka.

Untuk mengaktifkan akses tepercaya, Anda harus menjadi pengguna administrator di akun manajemen. Pengguna administrator adalah pengguna dengan izin penuh untuk Anda Akun AWS. Untuk informasi selengkapnya, [Buat pengguna administrator](#) di Panduan AWS Account Management

Referensi. Untuk rekomendasi untuk melindungi keamanan akun manajemen, lihat [Praktik terbaik untuk akun manajemen](#) di Panduan AWS Organizations Pengguna.

Untuk mengaktifkan akses tepercaya

1. Masuk AWS sebagai administrator akun manajemen dan buka CloudFormation konsol di <https://console.aws.amazon.com/cloudformation>.
2. Dari panel navigasi, pilih StackSets. Jika akses tepercaya dinonaktifkan, spanduk akan ditampilkan yang meminta Anda untuk mengaktifkan akses tepercaya.



3. Pilih Aktifkan akses tepercaya.

Akses tepercaya berhasil diaktifkan ketika banner berikut ditampilkan.



Note

Activate Organizations Access sama dengan Enable Organizations Access, dan Nonaktifkan Organizations Access sama dengan Disable Organizations Access. Ketentuan ini telah diperbarui berdasarkan pedoman pemasaran.

Untuk menonaktifkan akses tepercaya

Lihat [AWS CloudFormation StackSets dan AWS Organizations](#) di Panduan AWS Organizations Pengguna.

Sebelum Anda dapat menonaktifkan akses tepercaya dengan AWS Organizations, Anda harus membatalkan pendaftaran semua administrator yang didelegasikan. Untuk informasi selengkapnya, lihat [Mendaftarkan administrator yang didelegasikan](#).

Note

Untuk informasi tentang menggunakan operasi API alih-alih konsol untuk mengaktifkan atau menonaktifkan akses tepercaya, lihat:

- [ActivateOrganizationsAccess](#)
- [DeactivateOrganizationsAccess](#)
- [DescribeOrganizationsAccess](#)

Peran terkait layanan

Akun manajemen menggunakan peran

`AWSServiceRoleForCloudFormationStackSetsOrgAdmin` terkait layanan. Anda dapat mengubah atau menghapus peran ini hanya jika akses tepercaya dengan AWS Organizations dinonaktifkan.

Setiap akun target menggunakan peran

`AWSServiceRoleForCloudFormationStackSetsOrgMember` terkait layanan. Anda dapat mengubah atau menghapus peran ini hanya dalam dua kondisi: jika akses tepercaya dengan AWS Organizations dinonaktifkan, atau jika akun dihapus dari organisasi target atau unit organisasi (OU).

Untuk informasi selengkapnya, lihat [AWS CloudFormation StackSets dan AWS Organizations](#) di Panduan AWS Organizations Pengguna.

Daftarkan akun anggota administrator yang didelegasikan

Selain akun manajemen organisasi Anda, akun anggota dengan izin administrator yang didelegasikan dapat dibuat dan dikelola StackSets dengan izin yang dikelola layanan untuk organisasi. StackSets dengan izin yang dikelola layanan dibuat di akun manajemen, termasuk StackSets dibuat oleh administrator yang didelegasikan. Untuk terdaftar sebagai administrator yang didelegasikan untuk organisasi Anda, akun anggota Anda harus berada di organisasi. Untuk informasi selengkapnya tentang bergabung dengan organisasi, lihat [Mengundang Akun AWS untuk bergabung dengan organisasi Anda](#).

Organisasi Anda dapat memiliki hingga lima administrator yang didelegasikan terdaftar sekaligus. Administrator yang didelegasikan dapat memilih untuk menerapkan ke semua akun di organisasi atau spesifik Anda. OUs Akses tepercaya dengan AWS Organizations harus diaktifkan sebelum administrator yang didelegasikan dapat menyebarkan ke akun yang dikelola oleh Organizations. Untuk informasi selengkapnya, lihat [Aktifkan akses tepercaya untuk StackSets dengan AWS Organizations](#).

 Important

Harap perhatikan hal-hal berikut:

- Administrator yang didelegasikan memiliki izin penuh untuk men-deploy ke akun di organisasi Anda. Akun manajemen tidak dapat membatasi izin administrator yang didelegasikan untuk diterapkan ke operasi tertentu OUs atau untuk melakukan operasi tertentu. StackSet
- Pastikan administrator yang didelegasikan memiliki `organizations:ListDelegatedAdministrators` izin untuk menghindari kemungkinan kesalahan.

Anda dapat mendaftarkan administrator yang didelegasikan untuk organisasi Anda di Wilayah berikut: AS Timur (Ohio), AS Timur (Virginia N.), AS Barat (California N.), AS Barat (Oregon), Asia Pasifik (Mumbai), Asia Pasifik (Seoul), Asia Pasifik (Singapura), Asia Pasifik (Sydney), Asia Pasifik (Tokyo), Kanada (Tengah), Eropa (Frankfurt), Eropa (Irlandia) Irlandia, Eropa (London), Eropa (Paris), Eropa (Stockholm), Israel (Tel Aviv), Amerika Selatan (São Paulo), (AS-Timur), dan (AS-Barat). AWS GovCloud AWS GovCloud

[Anda dapat mendaftarkan dan membatalkan pendaftaran administrator yang didelegasikan menggunakan konsol,, atau.AWS CloudFormationAWS CLIAWS SDKs](#)

Untuk mendaftarkan administrator yang didelegasikan (konsol)

1. Masuk AWS sebagai administrator akun manajemen dan buka AWS CloudFormation konsol di<https://console.aws.amazon.com/cloudformation/>.
2. Dari panel navigasi, pilih StackSets.
3. Di bawah Administrator yang didelegasikan, pilih Daftarkan administrator yang didelegasikan.
4. Di kotak dialog Daftarkan administrator yang didelegasikan, pilih Daftarkan administrator yang didelegasikan.

Pesan berhasil menunjukkan bahwa akun anggota telah berhasil didaftarkan sebagai administrator yang didelegasikan.

Untuk membatalkan pendaftaran administrator yang didelegasikan (konsol)

1. Masuk AWS sebagai administrator akun manajemen dan buka AWS CloudFormation konsol di <https://console.aws.amazon.com/>.
2. Dari panel navigasi, pilih StackSets.
3. Di bawah Administrator yang didelegasikan, pilih akun yang ingin Anda batalkan pendaftarannya, lalu pilih Batalkan pendaftaran.

Pesan berhasil menunjukkan bahwa akun anggota telah berhasil dibatalkan pendaftarannya sebagai administrator yang didelegasikan.

Anda dapat mendaftarkan akun ini kembali kapan saja.

Untuk mendaftarkan administrator yang didelegasikan (AWS CLI)

1. Buka AWS CLI.
2. Jalankan perintah `register-delegated-administrator`.

```
$ aws organizations register-delegated-administrator \
  --service-principal=member.org.stacksets.cloudformation.amazonaws.com \
  --account-id="memberAccountId"
```

3. Jalankan perintah `list-delegated-administrators` untuk memverifikasi bahwa akun anggota tertentu berhasil terdaftar sebagai administrator didelegasikan.

```
$ aws organizations list-delegated-administrators \
  --service-principal=member.org.stacksets.cloudformation.amazonaws.com
```

Untuk membatalkan pendaftaran administrator yang didelegasikan (AWS CLI)

1. Buka AWS CLI.
2. Jalankan perintah `deregister-delegated-administrator`.

```
$ aws organizations deregister-delegated-administrator \
  --service-principal=member.org.stacksets.cloudformation.amazonaws.com \
  --account-id="memberAccountId"
```

3. Jalankan perintah `list-delegated-administrators` untuk memverifikasi bahwa akun anggota tertentu berhasil dibatalkan pendaftarannya sebagai administrator yang didelegasikan.

```
$ aws organizations list-delegated-administrators \
  --service-principal=member.org.stacksets.cloudformation.amazonaws.com
```

Anda dapat mendaftarkan akun ini kembali kapan saja.

Memulai dengan StackSets menggunakan template sampel

Tutorial ini akan membantu Anda memulai dengan StackSets menggunakan AWS Management Console. Ini memandu Anda melalui pembuatan StackSet menggunakan template sampel. Anda akan mempelajari cara menerapkan tumpukan di beberapa Wilayah, memantau StackSet operasi, dan melihat hasilnya.

Dalam tutorial ini, Anda akan membuat StackSet yang memungkinkan AWS Config di dalam Wilayah AS Barat (Oregon) (us-west-2) dan Wilayah AS Timur (Virginia N.) (us-east-1). Akun AWS Dengan StackSets, Anda dapat membuat, memperbarui, atau menghapus tumpukan di beberapa akun dan Wilayah dengan satu operasi, menjadikannya solusi ideal untuk mengelola infrastruktur dalam skala besar. Meskipun tutorial ini menggunakan satu akun untuk kesederhanaan, ini secara efektif menunjukkan kemampuan multi-wilayah. StackSets

Template sampel tersedia di bucket S3 berikut: <https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/AWSConfigAktifkan.yml.yml>.

Note

StackSets gratis, tetapi Anda akan dikenakan biaya untuk AWS sumber daya yang Anda buat dengannya, seperti AWS Config dalam tutorial ini. Untuk informasi selengkapnya, lihat [harga AWS Config](#).

Topik

- [Prasyarat](#)
- [Buat StackSet dengan contoh template dari konsol](#)
- [Pantau StackSet pembuatan](#)

- [Lihat StackSet hasil](#)
- [Perbarui StackSet](#)
- [Tambahkan tumpukan ke StackSet](#)
- [Bersihkan](#)
- [Langkah selanjutnya](#)

Prasyarat

Sebelum Anda memulai tutorial ini, pastikan Anda telah menyelesaikan prasyarat berikut:

- Anda harus menyiapkan peran IAM yang diperlukan untuk izin yang dikelola sendiri. Untuk membuat StackSet dan menyebarkan tumpukan dalam satu akun, Anda memerlukan peran berikut di akun Anda:
 - `AWSCloudFormationStackSetAdministrationRole`
 - `AWSCloudFormationStackSetExecutionRole`

Untuk petunjuk terperinci tentang pengaturan peran ini, lihat [Memberikan izin yang dikelola sendiri](#).

Buat StackSet dengan contoh template dari konsol

Untuk membuat StackSet yang memungkinkan AWS Config

1. Buka [konsol CloudFormation](#).
2. Pada bilah navigasi di bagian atas layar, pilih Wilayah AWS yang ingin Anda StackSet kelola.

Anda dapat memilih Wilayah mana pun yang mendukung StackSets. Wilayah yang Anda pilih tidak memengaruhi Wilayah mana yang dapat Anda StackSet gunakan.
3. Dari panel navigasi, pilih StackSets.
4. Dari bagian atas StackSetshalaman, pilih Buat StackSet.
5. Di bawah Izin, pilih Izin layanan mandiri dan pilih peran IAM yang Anda buat dalam prasyarat.
 - Untuk peran admin IAM, pilih `AWSCloudFormationStackSetAdministrationRole`.
 - Untuk nama peran eksekusi IAM, pilih `AWSCloudFormationStackSetExecutionRole`.
6. Dalam Prasyarat - Siapkan templat, pilih Gunakan templat sampel.
7. Di bawah Pilih templat sampel, pilih AWS Config template Aktifkan. Lalu, pilih Selanjutnya.

Template ini membuat sumber daya yang diperlukan untuk mengaktifkan AWS Config di akun Anda, termasuk perekam konfigurasi dan saluran pengiriman.

8. Pada halaman Tentukan StackSet detail, untuk StackSet nama, masukkan **my-awsconfig-stackset**.
9. Untuk StackSet deskripsi, masukkan **A StackSet that enables Config across multiple Regions**.
10. Di bawah Parameter, konfigurasi AWS Config pengaturan sebagai berikut:
 - a. Untuk Mendukung semua jenis sumber daya, pertahankan nilai default, true, untuk merekam semua jenis sumber daya yang didukung.
 - b. Untuk Sertakan tipe sumber daya global, pertahankan nilai default, false, untuk mengecualikan sumber daya global seperti peran IAM.
 - c. Tinggalkan Daftar jenis sumber daya jika tidak semua didukung disetel ke **<All>**.
 - d. Untuk wilayah yang berisi sumber daya peran terkait layanan Config, ganti **< >** dengan **DeployToAnyRegion us-west-2**

Ini berarti bahwa peran terkait layanan yang diberi nama hanya **AWSServiceRoleForConfig** akan dibuat jika tumpukan diarahkan ke Wilayah Barat AS (Oregon). Anda akan memilih Wilayah penerapan nanti dalam prosedur ini.

- e. Untuk frekuensi perekaman perekam konfigurasi, pilih perekaman **HARIAN**.
11. Pilih **Next** untuk melanjutkan.
12. Pada halaman **Configure StackSet options**, pilih **Add new tag** dan tambahkan tag dengan menentukan pasangan kunci dan nilai:
 - a. Untuk Kunci, masukkan **Stage**.
 - b. Untuk Nilai, masukkan **Test**.

Tag yang Anda terapkan StackSets diterapkan ke sumber daya yang dibuat oleh tumpukan Anda.

13. Untuk konfigurasi Eksekusi, pilih **Aktif** untuk mengaktifkan CloudFormation penanganan operasi yang dioptimalkan:
 - Operasi yang tidak bertentangan berjalan secara bersamaan untuk waktu penerapan yang lebih cepat.

- Operasi yang saling bertentangan secara otomatis diantrian dan diproses sesuai urutan yang diminta.

Saat operasi berjalan atau mengantri, CloudFormation mengantri semua operasi yang masuk meskipun tidak bertentangan. Anda tidak dapat mengubah pengaturan eksekusi selama waktu ini.

14. Pilih Berikutnya.
15. Pada halaman Setel opsi penerapan, untuk Tambahkan tumpukan StackSet, pilih Menerapkan tumpukan baru.
16. Untuk Akun, pilih Deploy tumpukan di akun.
17. Di kotak teks, masukkan Akun AWS ID Anda.
18. Untuk Tentukan wilayah, pilih Wilayah berikut dalam urutan ini:
 1. Wilayah Barat AS (Oregon) (us-west-2)
 2. Wilayah AS Timur (Virginia N.) (us-east-1)

Gunakan panah atas di sebelah Wilayah Barat AS (Oregon) untuk memindahkannya menjadi entri pertama dalam daftar jika diperlukan. Urutan Wilayah menentukan urutan penyebaran mereka.

19. Untuk opsi Deployment, konfigurasi pengaturan berikut:

- a. Untuk akun bersamaan maksimum, pertahankan default Nomor dan 1.

Untuk penerapan multi-akun, setelan ini berarti CloudFormation menyebarkan tumpukan Anda hanya dalam satu akun pada satu waktu.

- b. UntukToleransi kegagalan, jaga agar default tetap Angka dan 0.

Ini berarti bahwa maksimum penerapan nol tumpukan dapat gagal di salah satu Wilayah yang Anda tentukan sebelum CloudFormation menghentikan penerapan di Wilayah saat ini dan membatalkan penerapan di Wilayah yang tersisa.

- c. Untuk konkurensi Wilayah, pilih Sequential (default).

Pengaturan ini memastikan bahwa CloudFormation menyelesaikan penerapan di satu Wilayah sebelum pindah ke yang berikutnya.

- d. Untuk mode Konkurensi, pertahankan default toleransi kegagalan yang ketat.

Untuk penerapan multi-akun, ini mengurangi tingkat konkurensi akun saat kegagalan terjadi, tetap berada dalam toleransi Kegagalan +1.

20. Pilih Berikutnya.
21. Pada halaman Tinjau, tinjau pilihan Anda. Untuk membuat perubahan, pilih Edit pada bagian terkait.
22. Saat Anda siap untuk membuat StackSet, pilih Kirim.

Pantau StackSet pembuatan

Setelah Anda memilih Kirim, CloudFormation mulailah membuat StackSet dan menerapkan tumpukan ke Wilayah yang ditentukan di akun Anda. Halaman StackSet detail terbuka secara otomatis, di mana Anda dapat memantau kemajuan operasi.

Untuk memantau StackSet penciptaan

1. Pada halaman StackSet detail, tab Operasi ditampilkan secara default, menunjukkan operasi saat ini sedang berlangsung.
2. Status operasi harus RUNNING pada awalnya. CloudFormation membuat tumpukan di Wilayah yang Anda tentukan sesuai dengan opsi penerapan yang Anda konfigurasi.
3. Untuk melihat detail lebih lanjut tentang operasi, pilih ID operasi dalam daftar.
4. Pada halaman detail operasi, Anda dapat melihat status instance tumpukan yang dibuat di setiap Wilayah.
5. Tunggu status operasi berubah SUCCEEDED, yang menunjukkan bahwa StackSet dan semua instance tumpukannya berhasil dibuat.

Lihat StackSet hasil

Setelah StackSet pembuatan selesai, Anda dapat melihat instance tumpukan yang diterapkan dan memverifikasi bahwa AWS Config telah diaktifkan di akun Anda di seluruh Wilayah yang ditentukan.

Untuk melihat StackSet hasilnya

1. Pada halaman StackSet detail, pilih tab Stack instance.

2. Anda akan melihat daftar instance tumpukan yang dibuat di akun Anda di seluruh Wilayah yang ditentukan. Setiap instance stack harus memiliki status `SUCCEEDED`, yang menunjukkan bahwa itu berhasil diterapkan.
3. Untuk memverifikasi bahwa AWS Config diaktifkan di akun Anda, Anda dapat memeriksa AWS Config konsol di setiap Wilayah yang digunakan.

Perbarui StackSet

Setelah membuat StackSet, Anda mungkin ingin memperbaruinya untuk mengubah nilai parameter atau menambahkan lebih banyak Wilayah. Bagian ini menunjukkan cara memperbarui parameter frekuensi AWS Config perekaman.

Untuk memperbarui StackSet

1. Pada StackSet halaman, pilih **my-awsconfig-stackset**.
2. Dengan yang StackSet dipilih, pilih Edit StackSet detail dari menu Tindakan.
3. Pada halaman Pilih templat, untuk Prasyarat - Siapkan templat, pilih Gunakan templat saat ini.
4. Pilih Berikutnya.
5. Pada halaman Tentukan StackSet detail, di bawah Parameter, temukan Frekuensi perekaman perekam konfigurasi dan ubah dari `HARIAN` ke `KONTINU`.
6. Pilih Berikutnya.
7. Pada halaman Konfigurasi StackSet opsi, biarkan pengaturan apa adanya dan pilih Berikutnya.
8. Pada halaman Setel opsi penerapan, tentukan ID akun Anda dan Wilayah yang sama yang Anda gunakan saat membuat StackSet.
9. Untuk opsi Deployment, pertahankan pengaturan yang sama seperti sebelumnya.
10. Pilih Berikutnya.
11. Pada halaman Ulasan, tinjau perubahan Anda dan pilih Kirim.
12. CloudFormation mulai memperbarui Anda StackSet. Anda dapat memantau kemajuan pada tab Operasi pada halaman StackSet detail.

Tambahkan tumpukan ke StackSet

Anda dapat menambahkan lebih banyak tumpukan ke Anda StackSet dengan menerapkan ke Wilayah tambahan. Bagian ini menunjukkan cara menambahkan tumpukan ke Wilayah baru.

Untuk menambahkan tumpukan ke StackSet

1. Pada StackSetshalaman, pilih **my-awsconfig-stackset**.
2. Dengan yang StackSet dipilih, pilih Tambahkan tumpukan StackSet dari menu Tindakan.
3. Pada halaman Setel opsi penerapan, untuk Tambahkan tumpukan StackSet, pilih Menerapkan tumpukan baru.
4. Untuk Akun, pilih Menyebarkan tumpukan di akun dan masukkan ID akun Anda.
5. Untuk Tentukan wilayah, pilih Wilayah baru, seperti Eropa (Irlandia) eu-west-1 ().
6. Untuk opsi Deployment, pertahankan pengaturan yang sama seperti sebelumnya.
7. Pilih Berikutnya.
8. Pada halaman Specify Overrides, biarkan nilai properti seperti yang ditentukan dan pilih Berikutnya.
9. Pada halaman Tinjau, tinjau pilihan Anda dan pilih Kirim.
10. CloudFormation mulai membuat tumpukan baru di Wilayah yang ditentukan. Anda dapat memantau kemajuan pada tab Operasi pada halaman StackSet detail.

Bersihkan

Untuk menghindari biaya untuk AWS Config sumber daya yang tidak diinginkan, Anda harus membersihkan dengan menghapus tumpukan dari Anda StackSet, menghapus StackSet itu sendiri, dan menghapus peran IAM yang Anda buat untuk tutorial ini. Karena semua sumber daya digunakan dalam akun Anda, pembersihan sangat mudah.

Untuk menghapus tumpukan dari StackSet

1. Pada StackSetshalaman, pilih **my-awsconfig-stackset**.
2. Dengan yang StackSet dipilih, pilih Hapus tumpukan StackSet dari menu Tindakan.
3. Pada halaman Setel opsi penerapan, untuk Akun, pilih Menyebarkan tumpukan di akun dan masukkan ID akun Anda.
4. Untuk Tentukan wilayah, pilih semua Wilayah tempat Anda menerapkan tumpukan.
5. Untuk opsi Deployment, pertahankan pengaturan default.
6. Pastikan Pertahankan tumpukan tidak dihidupkan, sehingga tumpukan dan sumber dayanya dihapus.

7. Pilih Berikutnya.
8. Pada halaman Tinjau, tinjau pilihan Anda dan pilih Kirim.
9. CloudFormation mulai menghapus tumpukan dari Anda. StackSet Anda dapat memantau kemajuan pada tab Operasi pada halaman StackSet detail.

Untuk menghapus StackSet

1. Setelah semua tumpukan telah dihapus, pada StackSetshalaman, pilih **Andamy-awsconfig-stackset**.
2. Dengan yang StackSet dipilih, pilih Hapus StackSet dari menu Tindakan.
3. Saat diminta untuk mengonfirmasi, pilih Hapus.

Untuk menghapus peran layanan IAM

Karena Anda hanya menerapkan ke akun Anda, Anda hanya perlu menghapus peran IAM dari akun tunggal ini, membuat pembersihan jauh lebih sederhana daripada penerapan multi-akun.

1. Buka [konsol IAM](#).
2. Dari panel navigasi, pilih Peran.
3. Di kotak pencarian, masukkan **AWSCloudFormationStackSet** untuk menemukan peran yang Anda buat untuk tutorial ini.
4. Pilih kotak centang di sebelah **AWSCloudFormationStackSetAdministrationRole**.
5. Pilih Hapus dari bagian atas halaman.
6. Di kotak dialog konfirmasi, masukkan **delete** dan pilih Hapus.
7. Ulangi proses yang sama untuk menghapus file **AWSCloudFormationStackSetExecutionRole**.

Setelah menghapus StackSet, bucket Amazon S3 akan tetap ada di Wilayah AWS masing-masing karena atribut **DeletionPolicy** pada **AWS::S3::Bucket** sumber daya. Ini menyimpan data AWS Config riwayat Anda. Jika Anda tidak lagi membutuhkan data ini, Anda dapat menghapus ember dengan aman secara manual. Sebelum Anda dapat menghapus ember, Anda harus terlebih dahulu mengosongkannya. Mengosongkan ember menghapus semua objek di dalamnya.

Untuk mengosongkan dan menghapus ember Amazon S3

1. Buka [konsol Amazon S3](#).

2. Di panel navigasi di sisi kiri konsol, pilih Bucket.
3. Dalam daftar Bucket, Anda akan melihat bucket yang dibuat untuk ini StackSet di setiap Wilayah tempat Anda digunakan. Pilih opsi di sebelah nama bucket yang dibuat untuk ini StackSet, lalu pilih Empty.
4. Di halaman Bucket kosong, konfirmasi bahwa Anda ingin mengosongkan bucket dengan mengetik **permanently delete** ke dalam bidang teks, lalu pilih Kosongkan.
5. Pantau progres proses pengosongan bucket di halaman Bucket kosong: Status.
6. Untuk kembali ke daftar bucket Anda, pilih Keluar.
7. Pilih opsi di sebelah nama bucket, lalu pilih Hapus.
8. Saat diminta konfirmasi, ketikkan nama bucket lalu pilih Delete bucket.
9. Pantau perkembangan proses penghapusan bucket dari daftar Bucket. Saat Amazon S3 menyelesaikan penghapusan bucket, itu akan menghapus bucket dari daftar.
10. Ulangi proses ini untuk setiap bucket yang dibuat oleh StackSet di Wilayah yang berbeda.

Langkah selanjutnya

Selamat! Anda telah berhasil membuat template StackSet menggunakan contoh, menyebarkan tumpukan ke beberapa Wilayah dalam akun Anda, memperbarui, menambahkan lebih banyak tumpukan StackSet, dan membersihkan sumber daya Anda. Dengan berfokus pada penerapan akun tunggal, Anda telah menyederhanakan proses pembersihan sambil tetap mempelajari kemampuan inti multi-wilayah. StackSets

Untuk mempelajari lebih lanjut StackSets, jelajahi topik-topik berikut:

- [Ganti nilai parameter pada tumpukan di dalam CloudFormation StackSet](#)— Pelajari cara mengganti nilai parameter untuk akun dan Wilayah tertentu.
- [Buat AWS CloudFormation StackSets dengan izin yang dikelola layanan](#)— Jelajahi pembuatan StackSets untuk penerapan multi-akun dengan. AWS Organizations

Buat AWS CloudFormation StackSets dengan izin yang dikelola sendiri

Topik ini menjelaskan cara membuat StackSets dengan izin yang dikelola sendiri untuk menyebarkan tumpukan di seluruh dan Wilayah. Akun AWS

 Note

Sebelum melanjutkan, buat peran layanan IAM yang diperlukan StackSets untuk membangun hubungan tepercaya antara akun tempat Anda mengelola StackSet dari dan akun tempat Anda menerapkan tumpukan. Untuk informasi selengkapnya, lihat [Memberikan izin yang dikelola sendiri](#).

Topik

- [Buat StackSet dengan izin yang dikelola sendiri \(konsol\)](#)
- [Buat StackSet dengan izin yang dikelola sendiri \(\)AWS CLI](#)

Buat StackSet dengan izin yang dikelola sendiri (konsol)

Untuk membuat StackSet

1. Masuk ke AWS Management Console dan buka AWS CloudFormation konsol di <https://console.aws.amazon.com/cloudformation>.
2. Pada bilah navigasi di bagian atas layar, pilih Wilayah AWS yang ingin Anda StackSet kelola.
3. Dari panel navigasi, pilih StackSets.
4. Dari bagian atas StackSetshalaman, pilih Buat StackSet.
5. Di bawah Izin, pilih Izin layanan mandiri dan pilih peran IAM yang Anda buat.
6. Di bawah Prasyarat - Siapkan templat, pilih Template sudah siap.
7. Dalam Tentukan templat, pilih untuk menentukan URL untuk bucket S3 yang berisi templat tumpukan Anda atau unggah file templat tumpukan. Lalu, pilih Selanjutnya.
8. Pada halaman Tentukan StackSet detail, berikan nama untuk StackSet, tentukan parameter apa pun, lalu pilih Berikutnya.
9. Pilih Next untuk melanjutkan.
10. Pada halaman Konfigurasi StackSet opsi, di bawah Tag, tentukan tag apa pun yang akan diterapkan ke sumber daya di tumpukan Anda. Untuk informasi selengkapnya tentang cara tag digunakan AWS, lihat [Mengatur dan melacak AWS biaya menggunakan tag alokasi](#) biaya di Panduan AWS Manajemen Penagihan dan Biaya Pengguna.
11. Untuk konfigurasi Eksekusi, pilih Aktif untuk mengaktifkan CloudFormation penanganan operasi yang dioptimalkan:

- Operasi yang tidak bertentangan berjalan secara bersamaan untuk waktu penerapan yang lebih cepat.
- Operasi yang saling bertentangan secara otomatis diantri dan diproses sesuai urutan yang diminta.

Saat operasi berjalan atau mengantri, CloudFormation mengantri semua operasi yang masuk meskipun tidak bertentangan. Anda tidak dapat mengubah pengaturan eksekusi selama waktu ini.

12. Jika template Anda berisi sumber daya IAM, untuk Kemampuan, pilih Saya mengakui bahwa templat ini dapat membuat sumber daya IAM untuk menentukan bahwa Anda ingin menggunakan sumber daya IAM dalam templat. Untuk informasi selengkapnya, lihat [Mengakui sumber daya IAM dalam templat CloudFormation](#).
13. Pilih Berikutnya.
14. Pada halaman Setel opsi penerapan, untuk Tambahkan tumpukan StackSet, pilih Menerapkan tumpukan baru.
15. Untuk Akun, pilih Deploy tumpukan di akun. Tempel Akun AWS nomor target Anda di kotak teks, pisahkan beberapa angka dengan koma.

 Note

Anda dapat menyertakan ID akun administrator jika ingin menerapkan tumpukan di akun itu juga.

16. Di bawah Tentukan wilayah, pilih Wilayah tempat Anda ingin menerapkan tumpukan.
17. Untuk opsi Deployment, lakukan hal berikut:
 - Untuk Akun bersamaan maksimum, tentukan berapa banyak akun yang diproses secara bersamaan.
 - Untuk toleransi Kegagalan, tentukan jumlah maksimum kegagalan akun yang diizinkan per Wilayah. Operasi akan berhenti dan tidak akan melanjutkan ke Wilayah lain setelah batas ini tercapai.
 - Untuk konkurensi Wilayah, pilih cara memproses Wilayah: Berurutan (satu Wilayah pada satu waktu) atau Paralel (beberapa Wilayah secara bersamaan).
 - Untuk mode Konkurensi, pilih bagaimana konkurensi berperilaku selama eksekusi operasi.

- Toleransi kegagalan yang ketat — Mengurangi tingkat konkurensi akun saat kegagalan terjadi, tetap berada dalam toleransi Kegagalan +1.
- Toleransi kegagalan lunak - Mempertahankan tingkat konkurensi yang Anda tentukan (nilai akun bersamaan maksimum) terlepas dari kegagalan.

18. Pilih Berikutnya.

19. Pada halaman Tinjau, tinjau pilihan Anda. Untuk membuat perubahan, pilih Edit pada bagian terkait.

20. Saat Anda siap untuk membuat StackSet, pilih Kirim.

CloudFormation mulai membuat Anda StackSet. Lihat kemajuan dan status pembuatan tumpukan di halaman StackSet detail Anda StackSet yang terbuka saat Anda memilih Kirim.

Buat StackSet dengan izin yang dikelola sendiri ()AWS CLI

Ikuti langkah-langkah di bagian ini untuk menggunakan AWS CLI to:

- Buat StackSet wadahnya.
- Menyebarkan instance tumpukan.

Untuk membuat StackSet

1. Gunakan [create-stack-set](#) perintah untuk membuat StackSet nama baru *my-stackset*. Contoh berikut menggunakan template yang disimpan dalam bucket S3 dan menyertakan parameter *KeyPairName* yang menetapkan nilai *TestKey*.

```
aws cloudformation create-stack-set \  
  --stack-set-name my-stackset \  
  --template-url https://s3.region-code.amazonaws.com/amzn-s3-demo-bucket/  
MyApp.template \  
  --parameters ParameterKey=KeyPairName,ParameterValue=TestKey
```

2. Setelah create-stack-set perintah Anda selesai, jalankan [list-stack-sets](#) perintah untuk melihat bahwa Anda StackSet telah dibuat. Anda harus melihat yang baru StackSet dalam hasil.

```
aws cloudformation list-stack-sets
```

- Gunakan [create-stack-instances](#) perintah untuk menyebarkan tumpukan di dalam Anda. StackSet Contoh berikut menyebarkan tumpukan di dua Akun AWS (*account_ID_1* dan *account_ID_2*) di dua Wilayah (*us-west-2* dan *us-east-1*).

Tetapkan pemrosesan akun bersamaan dan preferensi penerapan lainnya menggunakan opsi. `--operation-preferences` Contoh ini menggunakan pengaturan berbasis hitungan. Perhatikan bahwa tidak `MaxConcurrentCount` boleh melebihi `FailureToleranceCount` + 1. Untuk pengaturan berbasis persentase, gunakan `FailureTolerancePercentage` atau `MaxConcurrentPercentage` sebagai gantinya.

```
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
```

Untuk informasi selengkapnya, lihat [CreateStackInstances](#) di dalam Referensi API AWS CloudFormation .

- Gunakan [describe-stack-set-operation](#) perintah untuk memverifikasi bahwa tumpukan Anda berhasil dibuat. Untuk `--operation-id` opsi, tentukan ID operasi yang dikembalikan sebagai bagian dari `create-stack-instances` output.

```
aws cloudformation describe-stack-set-operation \  
  --stack-set-name my-stackset \  
  --operation-id operation_ID
```

Buat AWS CloudFormation StackSets dengan izin yang dikelola layanan

Dengan izin yang dikelola layanan, Anda dapat menerapkan tumpukan ke akun yang dikelola oleh di Wilayah tertentu. AWS Organizations Dengan model ini, Anda tidak perlu membuat peran IAM yang diperlukan; CloudFormation membuat peran IAM atas nama Anda. Untuk informasi selengkapnya, lihat [Aktifkan akses tepercaya](#).

Topik

- [Pertimbangan](#)

- [Buat StackSet dengan izin yang dikelola layanan \(konsol\)](#)
- [Buat StackSet dengan izin yang dikelola layanan \(AWS CLI\)](#)

Pertimbangan

Sebelum Anda membuat izin yang dikelola layanan StackSet dengan layanan, pertimbangkan hal berikut:

- StackSets dengan izin yang dikelola layanan dibuat di akun manajemen, termasuk yang dibuat oleh administrator StackSets yang didelegasikan.
- Anda StackSet dapat menargetkan seluruh organisasi Anda (termasuk semua akun) atau unit organisasi tertentu (OUs). Jika Anda StackSet menargetkan OU orang tua, itu juga menargetkan anak mana pun OUs. Ketika StackSet target Anda spesifik OUs, semua akun di dalamnya OUs disertakan secara default. Namun, Anda dapat menargetkan akun tertentu menggunakan opsi filter akun.
- Beberapa StackSets dapat menargetkan organisasi atau OU yang sama.
- Anda tidak StackSet dapat menargetkan akun di luar organisasi Anda.
- Anda tidak StackSet dapat menerapkan tumpukan bersarang.
- CloudFormation tidak menyebarkan tumpukan ke akun manajemen organisasi, meskipun akun manajemen ada di organisasi Anda atau di OU di organisasi Anda.
- Penyebaran otomatis diatur pada StackSet level tersebut. Anda tidak dapat menyesuaikan penerapan otomatis secara selektif untuk OUs, akun, atau Wilayah.
- Izin entitas utama IAM (pengguna, peran, atau grup) yang Anda gunakan untuk masuk ke akun manajemen menentukan apakah Anda berwenang untuk menggunakan StackSets. Untuk contoh kebijakan IAM yang memberikan izin untuk diterapkan ke organisasi, lihat [Batasi operasi set tumpukan berdasarkan Wilayah dan tipe sumber daya](#)
- Administrator yang didelegasikan memiliki izin penuh untuk men-deploy ke akun di organisasi Anda. Akun manajemen tidak dapat membatasi izin administrator yang didelegasikan untuk diterapkan ke spesifik OUs atau untuk melakukan operasi tertentu. StackSet

Buat StackSet dengan izin yang dikelola layanan (konsol)

Untuk membuat StackSet

1. Masuk ke AWS Management Console dan buka AWS CloudFormation konsol di <https://console.aws.amazon.com/cloudformation>.
2. Pada bilah navigasi di bagian atas layar, pilih Wilayah AWS yang ingin Anda StackSet kelola.
3. Dari panel navigasi, pilih StackSets.
4. Dari bagian atas StackSetshalaman, pilih Buat StackSet.
5. Dalam Izin, pilih Izin yang dikelola layanan.

Note

Jika akses tepercaya dengan AWS Organizations dinonaktifkan, spanduk akan ditampilkan. Akses tepercaya diperlukan untuk membuat atau memperbarui izin StackSet yang dikelola layanan. Hanya administrator di akun manajemen organisasi yang memiliki izin untuk [Aktifkan akses tepercaya untuk StackSets dengan AWS Organizations](#).

6. Di bawah Prasyarat - Siapkan templat, pilih Template sudah siap.
7. Dalam Tentukan templat, pilih untuk menentukan URL untuk bucket S3 yang berisi templat tumpukan Anda atau unggah file templat tumpukan. Lalu, pilih Selanjutnya.
8. Pada halaman Tentukan StackSet detail, berikan nama untuk StackSet, tentukan parameter apa pun, lalu pilih Berikutnya.
9. Pada halaman Konfigurasi StackSet opsi, di bawah Tag, tentukan tag apa pun yang akan diterapkan ke sumber daya di tumpukan Anda. Untuk informasi selengkapnya tentang cara tag digunakan AWS, lihat [Mengatur dan melacak AWS biaya menggunakan tag alokasi](#) biaya di Panduan AWS Manajemen Penagihan dan Biaya Pengguna.
10. Untuk konfigurasi Eksekusi, pilih Aktif untuk mengaktifkan CloudFormation penanganan operasi yang dioptimalkan:
 - Operasi yang tidak bertentangan berjalan secara bersamaan untuk waktu penerapan yang lebih cepat.
 - Operasi yang saling bertentangan secara otomatis diantrian dan diproses sesuai urutan yang diminta.

Saat operasi berjalan atau mengantri, CloudFormation mengantri semua operasi yang masuk meskipun tidak bertentangan. Anda tidak dapat mengubah pengaturan eksekusi selama waktu ini.

11. Jika template Anda berisi sumber daya IAM, untuk Kemampuan, pilih Saya mengakui bahwa templat ini dapat membuat sumber daya IAM untuk menentukan bahwa Anda ingin menggunakan sumber daya IAM dalam templat. Untuk informasi selengkapnya, lihat [Mengakui sumber daya IAM dalam templat CloudFormation](#).
12. Pilih Berikutnya untuk melanjutkan dan mengaktifkan akses tepercaya jika belum diaktifkan.
13. Pada halaman Setel opsi penerapan, di bawah target Deployment, lakukan salah satu hal berikut:
 - Untuk menyebarkan ke semua akun di organisasi Anda, pilih Terapkan ke organisasi.
 - Untuk menyebarkan ke semua akun secara spesifik OUs, pilih Terapkan ke unit organisasi () OUs. Pilih Tambahkan OU, dan kemudian tempelkan ID OU target dalam kotak teks. Ulangi untuk setiap OU target baru.

Jika Anda memilih Deploy ke unit organisasi (OUs), untuk jenis filter Akun, Anda dapat mengatur target penerapan Anda menjadi akun individual tertentu dengan memilih salah satu opsi berikut dan memberikan nomor akun.

- Tidak ada (default) - Menerapkan tumpukan ke semua akun dalam yang ditentukan. OUs
 - Persimpangan — Menyebarkan tumpukan ke akun individu tertentu dalam yang dipilih. OUs
 - Perbedaan — Menyebarkan tumpukan ke semua akun di akun yang dipilih OUs kecuali untuk akun tertentu.
 - Union — Menyebarkan tumpukan ke akun individual tambahan yang ditentukan OUs ditambah.
14. Di bawah Penerapan otomatis, pilih apakah akan secara otomatis menerapkan ke akun yang ditambahkan ke organisasi target atau OUs di masa mendatang. Untuk informasi selengkapnya, lihat [Mengaktifkan atau menonaktifkan penerapan otomatis untuk in StackSets AWS Organizations](#).
 15. Jika Anda mengaktifkan deployment otomatis, dalam Perilaku penghapusan akun, pilih apakah sumber daya tumpukan dipertahankan atau dihapus ketika akun dihapus dari organisasi atau OU target.

 Note

Dengan tumpukan Pertahankan dipilih, tumpukan dihapus dari Anda StackSet, tetapi tumpukan dan sumber daya terkaitnya dipertahankan. Sumber daya tetap dalam keadaan mereka saat ini, tetapi tidak akan lagi menjadi bagian dari StackSet.

16. Di bawah Tentukan wilayah, pilih Wilayah tempat Anda ingin menerapkan tumpukan.
17. Untuk opsi Deployment, lakukan hal berikut:
 - Untuk Akun bersamaan maksimum, tentukan berapa banyak akun yang diproses secara bersamaan.
 - Untuk toleransi kegagalan, tentukan jumlah maksimum kegagalan akun yang diizinkan per Wilayah. Operasi akan berhenti dan tidak akan melanjutkan ke Wilayah lain setelah batas ini tercapai.
 - Untuk konkurensi Wilayah, pilih cara memproses Wilayah: Berurutan (satu Wilayah pada satu waktu) atau Paralel (beberapa Wilayah secara bersamaan).
 - Untuk mode Konkurensi, pilih bagaimana konkurensi berperilaku selama eksekusi operasi.
 - Toleransi kegagalan yang ketat — Mengurangi tingkat konkurensi akun saat kegagalan terjadi, tetap berada dalam toleransi Kegagalan +1.
 - Toleransi kegagalan lunak - Mempertahankan tingkat konkurensi yang Anda tentukan (nilai akun bersamaan maksimum) terlepas dari kegagalan.
18. Pilih Next untuk melanjutkan.
19. Pada halaman Tinjauan, verifikasi bahwa Anda StackSet akan menyebarkan ke akun yang benar di Wilayah yang benar, lalu pilih Buat StackSet.

Halaman StackSet detail terbuka. Anda dapat melihat kemajuan dan status pembuatan tumpukan di Anda StackSet.

Buat StackSet dengan izin yang dikelola layanan (AWS CLI)

Ikuti langkah-langkah di bagian ini untuk menggunakan AWS CLI to:

- Buat StackSet wadahnya.
- Menyebarkan instance tumpukan.

 Note

Saat bertindak sebagai administrator yang didelegasikan, Anda harus memasukkan `--call-as DELEGATED_ADMIN` dalam perintah.

Deploy to your organization

Untuk membuat StackSet

1. Gunakan [create-stack-set](#) perintah untuk membuat StackSet nama baru *my-stackset*. Contoh berikut menggunakan template yang disimpan dalam bucket S3, mengaktifkan penerapan otomatis, dan mempertahankan tumpukan saat akun dihapus. Untuk informasi selengkapnya, lihat [Mengaktifkan atau menonaktifkan penerapan otomatis untuk in StackSets AWS Organizations](#).

```
aws cloudformation create-stack-set \  
  --stack-set-name my-stackset \  
  --template-url https://s3.region-code.amazonaws.com/amzn-s3-demo-bucket/MyApp.template \  
  --permission-model SERVICE_MANAGED \  
  --auto-deployment Enabled=true,RetainStacksOnAccountRemoval=true
```

2. Gunakan [list-stack-sets](#) perintah untuk mengonfirmasi bahwa Anda StackSet telah dibuat. StackSet Yang baru Anda tercantum dalam hasil.

```
aws cloudformation list-stack-sets
```

- Jika Anda menetapkan `--call-as` opsi untuk `DELEGATED_ADMIN` saat masuk ke akun anggota Anda, `list-stack-sets` mengembalikan semua StackSets dengan izin yang dikelola layanan di akun manajemen organisasi.
 - Jika Anda menyetel `--call-as` opsi untuk `SELF` saat masuk Akun AWS, `list-stack-sets` kembalikan semua yang dikelola sendiri StackSets di Anda Akun AWS.
 - Jika Anda menetapkan `--call-as` opsi `SELF` saat masuk ke akun manajemen organisasi, `list-stack-sets` mengembalikan semua yang ada StackSets di akun manajemen organisasi.
3. Gunakan [create-stack-instances](#) perintah untuk menambahkan tumpukan ke Anda StackSet. Untuk `--deployment-targets` opsi, tentukan ID root organisasi yang akan diterapkan ke semua akun di organisasi Anda.

Tetapkan pemrosesan akun bersamaan dan preferensi penerapan lainnya menggunakan opsi. `--operation-preferences` Contoh ini menggunakan pengaturan berbasis hitungan. Perhatikan bahwa tidak `MaxConcurrentCount` boleh melebihi `FailureToleranceCount` + 1. Untuk pengaturan berbasis persentase, gunakan `FailureTolerancePercentage` atau `MaxConcurrentPercentage` sebagai gantinya.

```
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
```

Untuk informasi selengkapnya, lihat [CreateStackInstances](#) di dalam Referensi API AWS CloudFormation .

4. Menggunakan `operation-id` yang dikembalikan sebagai bagian dari `create-stack-instances` output, gunakan [describe-stack-set-operation](#) perintah berikut untuk memverifikasi bahwa tumpukan Anda berhasil dibuat.

```
aws cloudformation describe-stack-set-operation \
--stack-set-name my-stackset \
--operation-id operation_ID
```

Deploy to organizational units (OUs)

Untuk membuat StackSet

1. Gunakan [create-stack-set](#) perintah untuk membuat StackSet nama baru *my-stackset*. Contoh berikut menggunakan template yang disimpan dalam bucket S3 dan menyertakan parameter *KeyPairName* yang menetapkan nilai *TestKey*

```
aws cloudformation create-stack-set \
--stack-set-name my-stackset \
--template-url https://s3.region-code.amazonaws.com/amzn-s3-demo-bucket/MyApp.template \
--permission-model SERVICE_MANAGED \
--parameters ParameterKey=KeyPairName,ParameterValue=TestKey
```

2. Gunakan [list-stack-sets](#) perintah untuk mengonfirmasi bahwa Anda StackSet telah dibuat. StackSet Yang baru Anda tercantum dalam hasil.

```
aws cloudformation list-stack-sets
```

- Jika Anda menetapkan `--call-as` opsi untuk `DELEGATED_ADMIN` saat masuk ke akun anggota Anda, `list-stack-sets` mengembalikan semua StackSets dengan izin yang dikelola layanan di akun manajemen organisasi.
 - Jika Anda menyetel `--call-as` opsi untuk `SELF` saat masuk Akun AWS, `list-stack-sets` kembalikan semua yang dikelola sendiri StackSets di Anda Akun AWS.
 - Jika Anda menetapkan `--call-as` opsi `SELF` saat masuk ke akun manajemen organisasi, `list-stack-sets` mengembalikan semua yang ada StackSets di akun manajemen organisasi.
3. Gunakan [create-stack-instances](#) perintah untuk menambahkan tumpukan ke Anda StackSet. Untuk `--deployment-targets` opsi, tentukan OU IDs yang akan digunakan.

Tetapkan pemrosesan akun bersamaan dan preferensi penerapan lainnya menggunakan opsi. `--operation-preferences` Contoh ini menggunakan pengaturan berbasis hitungan. Perhatikan bahwa tidak `MaxConcurrentCount` boleh melebihi `FailureToleranceCount` + 1. Untuk pengaturan berbasis persentase, gunakan `FailureTolerancePercentage` atau `MaxConcurrentPercentage` sebagai gantinya.

```
aws cloudformation create-stack-instances --stack-set-name my-stackset \  
  --deployment-targets OrganizationalUnitIds=ou-rcuk-1x5j1lwo,ou-rcuk-slrl5lh0a \  
  --regions us-west-2 us-east-1 \  
  --operation-preferences MaxConcurrentCount=1,FailureToleranceCount=0
```

Untuk informasi selengkapnya, lihat [CreateStackInstances](#) di dalam Referensi API AWS CloudFormation .

4. Menggunakan `operation-id` yang dikembalikan sebagai bagian dari `create-stack-instances` output, gunakan [describe-stack-set-operation](#) perintah berikut untuk memverifikasi bahwa tumpukan Anda berhasil dibuat.

```
aws cloudformation describe-stack-set-operation \  
  --stack-set-name my-stackset \  
  --operation-id operation_ID
```

Deploy to specific accounts in OUs

Anda dapat menargetkan unit organisasi tertentu (OUs) dan menggunakan pemfilteran akun untuk secara tepat mengontrol akun mana yang menerima penerapan tumpukan. Secara default, tumpukan disebar ke semua akun dalam yang ditentukan OUs jika tidak ada pemfilteran akun yang ditentukan.

Di AWS CLI, Anda menentukan pemfilteran akun dengan `--deployment-targets` opsi. Untuk informasi selengkapnya, lihat [DeploymentTargets](#).

Setelah Anda membuat StackSet wadah dengan `create-stack-set` perintah, gunakan salah satu contoh berikut untuk menyebarkan tumpukan ke akun tertentu.

Targetkan akun tertentu dalam OU

Contoh berikut menyebarkan tumpukan hanya ke akun A1 dan A2 di 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
```

Kecualikan akun dari OU

Contoh berikut menyebarkan tumpukan ke semua akun OU1 kecuali akun A1 dan A2.

```
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
```

Sertakan akun tambahan

Contoh berikut menyebarkan tumpukan ke semua akun di OU1 plus akun tambahan A4 dari OU lain.

```
aws cloudformation create-stack-instances --stack-set-name my-stackset \  
  --deployment-targets OrganizationalUnitIds=OU1,Accounts=A4,AccountFilterType=UNION  
  \  
  --regions us-west-2 us-east-1
```

Mengaktifkan atau menonaktifkan penerapan otomatis untuk in StackSets AWS Organizations

CloudFormation dapat secara otomatis menerapkan tumpukan tambahan ke AWS Organizations akun baru saat ditambahkan ke organisasi target atau unit organisasi (OU). Anda dapat mengaktifkan penerapan otomatis dan memilih apakah akan menghapus atau mempertahankan tumpukan dan sumber daya terkait saat akun dihapus dari target. OUs Pengaturan ini dapat dimodifikasi kapan saja untuk menggunakan izin StackSets yang dikelola layanan.

Cara kerja penerapan otomatis

Ketika penerapan otomatis diaktifkan, mereka dipicu ketika akun ditambahkan ke organisasi target atau OU, dihapus dari organisasi target atau OU, atau dipindahkan di antara target. OUs

Misalnya, pertimbangkan StackSet1 target OU1 di us-east-1 Wilayah dan target StackSet2 itu OU2 di us-east-1 Wilayah. OU1 berisi AccountA.

Jika Anda beralih AccountA dari OU1 ke OU2 dengan penerapan otomatis diaktifkan, CloudFormation secara otomatis menjalankan operasi penghapusan untuk menghapus StackSet1 tumpukan dari AccountA dan mengantri operasi buat yang menambahkan tumpukan ke StackSet2 AccountA

Pertimbangan

Berikut ini adalah pertimbangan saat menggunakan penerapan otomatis:

- Fitur penerapan otomatis diaktifkan di level tersebut. StackSet Anda tidak dapat menyesuaikan penerapan otomatis secara selektif untuk OUs, akun, atau Wilayah.
- Nilai parameter yang diganti hanya berlaku untuk akun yang saat ini ada di target OUs dan anak mereka. OUs Akun yang ditambahkan ke target OUs dan anak mereka OUs di masa mendatang akan menggunakan nilai StackSet default dan bukan nilai yang diganti.
- Jika Anda menargetkan akun tertentu dan mengaktifkan penerapan otomatis, Anda StackSet akan terus menggunakan filter tingkat akun yang ditentukan dalam penerapan terakhir sementara juga menerapkan ke akun yang baru ditambahkan dalam organisasi yang diterapkan. Untuk mencegah penyebaran ke akun yang baru ditambahkan, nonaktifkan penerapan otomatis.

Mengaktifkan atau menonaktifkan penerapan otomatis (konsol)

Untuk mengaktifkan atau menonaktifkan penerapan otomatis

1. Masuk ke AWS Management Console dan buka AWS CloudFormation konsol di <https://console.aws.amazon.com/cloudformation>.
2. Pada bilah navigasi di bagian atas layar, pilih yang Wilayah AWS Anda buat StackSet di.
3. Dari panel navigasi, pilih StackSets.
4. Pada StackSetshalaman, pilih opsi di sebelah nama StackSet untuk memperbarui.
5. Pilih Edit penerapan otomatis dari menu Tindakan di sudut kanan atas.
6. Dari kotak dialog yang terbuka, lakukan hal berikut:
 - a. Untuk penerapan Otomatis, pilih Diaktifkan atau Dinonaktifkan.
 - b. Untuk perilaku penghapusan Akun, pilih Hapus tumpukan atau Pertahankan tumpukan. Sumber daya yang dipertahankan tetap dalam keadaan mereka saat ini, tetapi tidak akan lagi menjadi bagian dari StackSet
7. Pilih Simpan.

Mengaktifkan atau menonaktifkan penerapan otomatis ()AWS CLI

Untuk mengaktifkan atau menonaktifkan penerapan otomatis

1. Gunakan perintah [update-stack-set](#) dengan opsi `--auto-deployment`.

Perintah berikut memungkinkan penerapan otomatis.

```
aws cloudformation update-stack-set --stack-set-name my-stackset \  
  --use-previous-template --auto-deployment  
  Enabled=true,RetainStacksOnAccountRemoval=true
```

Atau, untuk menonaktifkan penerapan otomatis, tentukan `Enabled=false` sebagai nilai untuk `--auto-deployment` opsi, seperti pada contoh berikut.

```
aws cloudformation update-stack-set --stack-set-name my-stackset \  
  --use-previous-template --auto-deployment Enabled=false
```

2. Menggunakan ID operasi yang dikembalikan sebagai bagian dari update-stack-set output, jalankan [describe-stack-set-operation](#) untuk memverifikasi bahwa Anda StackSet telah berhasil diperbarui.

```
aws cloudformation describe-stack-set-operation --operation-id operation_ID
```

Perbarui AWS CloudFormation StackSets

Anda dapat memperbarui Anda StackSet menggunakan CloudFormation konsol atau AWS CLI.

Untuk menambah dan menghapus akun dan Wilayah dari StackSet, lihat [Tambahkan tumpukan ke StackSets](#) dan [Hapus tumpukan dari StackSets](#). Untuk mengganti nilai parameter untuk tumpukan, lihat [Ganti parameter pada tumpukan](#).

Topik

- [Perbarui StackSet \(konsol\)](#)
- [Perbarui StackSet \(AWS CLI\)](#)

Perbarui StackSet (konsol)

Untuk memperbarui StackSet

1. Masuk ke AWS Management Console dan buka AWS CloudFormation konsol di <https://console.aws.amazon.com/cloudformation>.
2. Pada bilah navigasi di bagian atas layar, pilih yang Wilayah AWS Anda buat StackSet di.
3. Dari panel navigasi, pilih StackSets.
4. Pada StackSetshalaman, pilih yang ingin StackSet Anda perbarui.
5. Dengan yang StackSet dipilih, pilih Edit StackSet detail dari menu Tindakan.
6. Pada halaman Pilih templat, perbarui bagian Izin sesuai kebutuhan, atau lewati ke langkah berikutnya.
7. Untuk Prasyarat - Siapkan templat, pilih Gunakan templat saat ini untuk menggunakan templat saat ini, atau Ganti templat saat ini untuk menentukan URL S3 ke templat lain atau unggah templat baru.
8. Pilih Berikutnya.

9. Pada halaman Tentukan StackSet detail, untuk StackSet deskripsi, perbarui deskripsi StackSet sesuai kebutuhan.
10. Untuk Parameter, perbarui nilai parameter sesuai kebutuhan.
11. Pilih Berikutnya.
12. Pada halaman StackSet opsi Konfigurasi, untuk Tag, ubah tag sesuai kebutuhan. Anda dapat menambahkan, memperbarui, atau menghapus tag. Untuk informasi selengkapnya tentang cara tag digunakan AWS, lihat [Mengatur dan melacak AWS biaya menggunakan tag alokasi](#) biaya di Panduan AWS Manajemen Penagihan dan Biaya Pengguna.
13. Untuk konfigurasi Eksekusi, Anda dapat memperbarui konfigurasi eksekusi sesuai kebutuhan.

 Note

Ingat, Anda tidak dapat mengubah pengaturan eksekusi saat operasi sedang berjalan atau mengantri.

14. Jika template Anda berisi sumber daya IAM, untuk Kemampuan, pilih Saya mengakui bahwa templat ini dapat membuat sumber daya IAM untuk menentukan bahwa Anda ingin menggunakan sumber daya IAM dalam templat. Untuk informasi selengkapnya, lihat [Mengakui sumber daya IAM dalam templat CloudFormation](#).
15. Pilih Berikutnya.
16. Pada halaman Setel opsi penyebaran, berikan akun dan Wilayah untuk pembaruan.

CloudFormation akan menerapkan pembaruan tumpukan di akun yang ditentukan dalam Wilayah pertama, lalu beralih ke yang berikutnya, dan seterusnya, selama kegagalan penerapan Wilayah tidak melebihi toleransi kegagalan yang ditentukan.

- a. [Izin yang dikelola sendiri] Untuk Akun, lokasi Penerapan, pilih Terapkan tumpukan di akun. Rekatkan akun target IDs yang Anda gunakan untuk membuat StackSet di kotak teks, pisahkan beberapa angka dengan koma.

[Izin yang dikelola layanan] Lakukan salah satu hal berikut:

- Pilih Menyebarkan ke unit organisasi (OUs). Masukkan target OUs yang Anda gunakan untuk membuat StackSet.
- Pilih Deploy ke akun. Rekatkan OU target IDs atau akun IDs yang Anda gunakan untuk membuat StackSet.

- b. Untuk Tentukan wilayah, tentukan urutan pemutakhiran yang CloudFormation ingin Anda gunakan.
 - c. Untuk opsi Deployment, lakukan hal berikut:
 - Untuk Akun bersamaan maksimum, tentukan berapa banyak akun yang diproses secara bersamaan.
 - Untuk toleransi Kegagalan, tentukan jumlah maksimum kegagalan akun yang diizinkan per Wilayah. Operasi akan berhenti dan tidak akan melanjutkan ke Wilayah lain setelah batas ini tercapai.
 - Untuk konkurensi Wilayah, pilih cara memproses Wilayah: Berurutan (satu Wilayah pada satu waktu) atau Paralel (beberapa Wilayah secara bersamaan).
 - Untuk mode Konkurensi, pilih bagaimana konkurensi berperilaku selama eksekusi operasi.
 - Toleransi kegagalan yang ketat — Mengurangi tingkat konkurensi akun saat kegagalan terjadi, tetap berada dalam toleransi Kegagalan +1.
 - Toleransi kegagalan lunak - Mempertahankan tingkat konkurensi yang Anda tentukan (nilai akun bersamaan maksimum) terlepas dari kegagalan.
 - d. Pilih Next untuk melanjutkan.
17. Pada halaman Tinjau, tinjau pilihan Anda. Untuk membuat perubahan, pilih Edit pada bagian terkait.
18. Saat Anda siap untuk melanjutkan, pilih Kirim.

CloudFormation mulai menerapkan pembaruan Anda ke Anda StackSet, dan menampilkan tab Operasi pada halaman StackSet detail. Anda dapat melihat kemajuan dan status operasi pembaruan pada tab Operasi.

Perbarui StackSet (AWS CLI)

Note

Saat bertindak sebagai administrator yang didelegasikan, Anda harus memasukkan `--call-as DELEGATED_ADMIN` dalam perintah.

1. Untuk memperbarui StackSet

Gunakan [update-stack-set](#) perintah untuk membuat perubahan pada Anda StackSet.

Dalam contoh berikut, kami memperbarui `--parameters` opsi StackSet by using. Secara khusus, kami mengubah frekuensi pengiriman snapshot default untuk konfigurasi saluran pengiriman dari `TwentyFour_Hours` ke `Twelve_Hours`. Karena kita masih menggunakan template saat ini, kita menambahkan `--use-previous-template` opsi.

Tetapkan pemrosesan akun bersamaan dan preferensi penerapan lainnya menggunakan opsi. `--operation-preferences` Contoh-contoh ini menggunakan pengaturan berbasis hitungan. Perhatikan bahwa tidak `MaxConcurrentCount` boleh melebihi `FailureToleranceCount` + 1. Untuk pengaturan berbasis persentase, gunakan `FailureTolerancePercentage` atau `MaxConcurrentPercentage` sebagai gantinya.

[Izin yang dikelola sendiri] Untuk `--accounts` opsi ini, berikan akun yang ingin IDs Anda targetkan pembaruan.

```
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
```

[Izin yang dikelola layanan] Untuk `--deployment-targets` opsi ini, berikan ID root organisasi atau unit organisasi (OU) yang ingin IDs Anda targetkan pembaruan.

```
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
```

Untuk informasi selengkapnya, lihat [UpdateStackSet](#) di dalam Referensi API AWS CloudFormation .

2. Verifikasi StackSet bahwa Anda telah berhasil diperbarui dengan menjalankan `describe-stack-set-operation` perintah untuk menampilkan status dan hasil operasi pembaruan Anda. Untuk `--operation-id`, gunakan ID operasi yang dikembalikan oleh perintah `update-stack-set` Anda.

```
aws cloudformation describe-stack-set-operation \  
--operation-id operation_ID
```

Tambahkan tumpukan ke CloudFormation StackSets

Saat Anda membuat StackSet, Anda dapat membuat tumpukan untuk itu StackSet. CloudFormation juga memungkinkan Anda untuk menambahkan lebih banyak tumpukan, untuk akun dan Wilayah tambahan, kapan saja setelah StackSet dibuat. Anda dapat menambahkan tumpukan menggunakan CloudFormation konsol atau. AWS CLI

Topik

- [Tambahkan tumpukan ke StackSet \(konsol\)](#)
- [Tambahkan tumpukan ke StackSet \(AWS CLI\)](#)

Tambahkan tumpukan ke StackSet (konsol)

Untuk menambahkan tumpukan ke StackSet

1. Masuk ke AWS Management Console dan buka AWS CloudFormation konsol di <https://console.aws.amazon.com/cloudformation>.
2. Pada bilah navigasi di bagian atas layar, pilih yang Wilayah AWS Anda buat StackSet di.
3. Dari panel navigasi, pilih StackSets. Pada StackSets halaman, pilih StackSet yang Anda buat.
4. Dengan yang StackSet dipilih, pilih Tambahkan tumpukan StackSet dari menu Tindakan.
5. Pada halaman Set deployment options, lakukan hal berikut:
 - a. Untuk Tambahkan tumpukan StackSet, pilih Menerapkan tumpukan baru.
 - b. Selanjutnya, lakukan hal berikut tergantung pada konfigurasi StackSet izin Anda:
 - [Izin yang dikelola sendiri] Untuk Akun, lokasi Penerapan, pilih Terapkan tumpukan di akun. Tempelkan angka akun target Anda di kotak teks, pisahkan beberapa angka dengan koma.

- [Izin yang dikelola layanan] Untuk target Deployment, lakukan salah satu hal berikut:
 - Pilih Deploy ke organisasi untuk men-deploy ke semua akun di organisasi Anda.
 - Pilih Deploy to organization units (OUs) untuk menyebarkan ke semua akun secara spesifik. OUs Pilih Tambahkan OU lain, dan kemudian tempelkan ID OU target dalam kotak teks. Ulangi untuk setiap target baru OU. CloudFormation juga menargetkan setiap anak OUs dari target yang Anda pilih.

 Note

Jika Anda menambahkan OU yang StackSet sudah Anda targetkan, CloudFormation buat tumpukan baru di akun apa pun di OU yang belum memiliki tumpukan dari Anda StackSet (misalnya, akun yang ditambahkan ke OU setelah Anda StackSet dibuat dan dengan penerapan otomatis dinonaktifkan).

- c. Untuk Tentukan wilayah, tentukan wilayah mana yang Wilayah AWS akan digunakan di akun target yang Anda tentukan di langkah sebelumnya. Secara default, CloudFormation akan menerapkan tumpukan di akun yang ditentukan dalam Wilayah pertama, lalu beralih ke yang berikutnya, dan seterusnya, selama kegagalan penerapan Wilayah tidak melebihi toleransi kegagalan yang ditentukan.
- d. Untuk opsi Deployment, lakukan hal berikut:
 - Untuk Akun bersamaan maksimum, tentukan berapa banyak akun yang diproses secara bersamaan.
 - Untuk toleransi Kegagalan, tentukan jumlah maksimum kegagalan akun yang diizinkan per Wilayah. Operasi akan berhenti dan tidak akan melanjutkan ke Wilayah lain setelah batas ini tercapai.
 - Untuk konkurensi Wilayah, pilih cara memproses Wilayah: Berurutan (satu Wilayah pada satu waktu) atau Paralel (beberapa Wilayah secara bersamaan).
 - Untuk mode Konkurensi, pilih bagaimana konkurensi berperilaku selama eksekusi operasi.
 - Toleransi kegagalan yang ketat — Mengurangi tingkat konkurensi akun saat kegagalan terjadi, tetap berada dalam toleransi Kegagalan +1.
 - Toleransi kegagalan lunak - Mempertahankan tingkat konkurensi yang Anda tentukan (nilai akun bersamaan maksimum) terlepas dari kegagalan.
- e. Pilih Berikutnya.

6. Pada halaman Tentukan Penimpaan, biarkan nilai properti seperti yang ditentukan. Anda tidak akan menimpa nilai properti untuk tumpukan yang akan Anda buat. Pilih Berikutnya.
7. Pada halaman Tinjau, tinjau pilihan Anda. Untuk membuat perubahan, pilih Edit pada bagian terkait.
8. Saat Anda siap untuk melanjutkan, pilih Kirim.

CloudFormation mulai membuat tumpukan Anda. Lihat kemajuan dan status pembuatan tumpukan di halaman StackSet detail Anda StackSet yang terbuka saat Anda memilih Kirim. Setelah selesai, tumpukan baru Anda harus terdaftar di tab Stack instance.

Tambahkan tumpukan ke StackSet (AWS CLI)

Note

Saat bertindak sebagai administrator yang didelegasikan, Anda harus memasukkan `--call-as DELEGATED_ADMIN` dalam perintah.

Untuk menambahkan tumpukan ke StackSet dengan izin yang dikelola sendiri

Gunakan perintah CLI `create-stack-instances`. Untuk `--accounts` opsi, berikan akun IDs yang ingin Anda buat tumpukan.

```
aws cloudformation create-stack-instances --stack-set-name my-stackset \  
--accounts account_id --regions eu-west-1 us-west-2
```

Untuk menambahkan tumpukan ke StackSet dengan izin yang dikelola layanan

Gunakan perintah CLI `create-stack-instances`. Untuk `--deployment-targets` opsi, berikan ID organisasi (root) atau OU IDs yang ingin Anda buat tumpukan. Misalnya perintah yang menargetkan akun tertentu, lihat [Buat StackSet dengan izin yang dikelola layanan \(AWS CLI\)](#).

```
aws cloudformation create-stack-instances --stack-set-name my-stackset \  
--deployment-targets OrganizationalUnitIds=ou-rcuk-r1qi0wl7 --regions eu-west-1 us-west-2
```


 Note

Jika Anda menambahkan OU yang StackSet sudah Anda targetkan, CloudFormation buat tumpukan baru di akun apa pun di OU yang belum memiliki tumpukan dari Anda StackSet (misalnya, akun yang ditambahkan ke OU setelah Anda StackSet dibuat dan dengan penerapan otomatis dinonaktifkan).

Ganti nilai parameter pada tumpukan di dalam CloudFormation StackSet

Dalam kasus tertentu, Anda mungkin ingin tumpukan di Wilayah atau akun tertentu memiliki nilai properti yang berbeda dari yang ditentukan dalam StackSet dirinya sendiri. Misalnya, Anda mungkin ingin menentukan nilai yang berbeda untuk parameter yang diberikan berdasarkan apakah akun digunakan untuk pengembangan atau produksi. Untuk situasi ini, CloudFormation Anda dapat mengganti nilai parameter dalam tumpukan berdasarkan akun dan Wilayah. Anda dapat mengganti nilai parameter template saat pertama kali membuat tumpukan, dan Anda dapat mengganti nilai parameter untuk tumpukan yang ada. Anda hanya dapat mengatur parameter yang sebelumnya telah diganti dalam tumpukan kembali ke nilai yang ditentukan dalam StackSet

Penggantian nilai parameter berlaku untuk tumpukan di akun dan Wilayah yang Anda pilih. Selama StackSet pembaruan, nilai parameter apa pun yang diganti untuk tumpukan tidak diperbarui, tetapi mempertahankan nilai yang diganti.

Anda hanya dapat mengganti nilai parameter yang ditentukan dalam StackSet; untuk menambah atau menghapus parameter itu sendiri, Anda perlu memperbarui StackSet template. Jika Anda menambahkan parameter ke StackSet template, maka sebelum Anda dapat mengganti nilai parameter itu dalam tumpukan, Anda harus terlebih dahulu memperbarui semua tumpukan dengan parameter dan nilai baru yang ditentukan dalam StackSet. Setelah semua tumpukan diperbarui dengan parameter baru, Anda kemudian dapat mengganti nilai parameter di masing-masing tumpukan sesuai keinginan.

Untuk mempelajari cara mengganti nilai StackSet parameter saat Anda membuat tumpukan, lihat.

[Tambahkan tumpukan ke StackSets](#)

Topik

- [Ganti parameter pada tumpukan \(konsol\)](#)

- [Ganti parameter pada tumpukan \(\)AWS CLI](#)

Ganti parameter pada tumpukan (konsol)

Untuk mengganti parameter untuk tumpukan tertentu

1. Masuk ke AWS Management Console dan buka AWS CloudFormation konsol di <https://console.aws.amazon.com/cloudformation>.
2. Pada bilah navigasi di bagian atas layar, pilih yang Wilayah AWS Anda buat StackSet di.
3. Dari panel navigasi, pilih StackSets. Pada StackSets halaman, pilih StackSet.
4. Dengan yang StackSet dipilih, pilih Ganti StackSet parameter dari menu Tindakan.
5. Pada halaman Setel opsi penerapan, berikan akun dan Wilayah untuk tumpukan yang akan Anda buat penggantian.

Secara default, CloudFormation akan menerapkan tumpukan di akun yang ditentukan dalam Wilayah pertama, lalu beralih ke yang berikutnya, dan seterusnya, asalkan kegagalan penerapan Wilayah tidak melebihi toleransi kegagalan yang ditentukan.

- a. [Izin yang dikelola sendiri] Untuk lokasi Deployment, pilih Menerapkan tumpukan di akun. Tempel beberapa atau semua akun target IDs yang Anda gunakan untuk membuat akun Anda StackSet.

[Izin yang dikelola layanan] Lakukan salah satu hal berikut:

- Pilih Menyebar ke unit organisasi (OUs). Masukkan satu atau lebih target OUs yang Anda gunakan untuk membuat StackSet. Nilai parameter yang diganti hanya berlaku untuk akun yang saat ini ada di target OUs dan anak mereka. OUs Akun yang ditambahkan ke target OUs dan anak mereka OUs di masa mendatang akan menggunakan nilai StackSet default dan bukan nilai yang diganti.
 - Pilih Deploy ke akun. Tempelkan beberapa atau semua target OU IDs atau akun IDs yang Anda gunakan untuk membuat StackSet.
- b. Untuk Tentukan wilayah, tambahkan satu atau beberapa Wilayah tempat Anda telah menerapkan tumpukan untuk ini. StackSet

Jika Anda menambahkan beberapa Wilayah, urutan Wilayah dalam Tentukan wilayah menentukan urutan deployment mereka.

- c. Untuk opsi Deployment, lakukan hal berikut:

- Untuk Akun bersamaan maksimum, tentukan berapa banyak akun yang diproses secara bersamaan.
- Untuk toleransi Kegagalan, tentukan jumlah maksimum kegagalan akun yang diizinkan per Wilayah. Operasi akan berhenti dan tidak akan melanjutkan ke Wilayah lain setelah batas ini tercapai.
- Untuk konkurensi Wilayah, pilih cara memproses Wilayah: Berurutan (satu Wilayah pada satu waktu) atau Paralel (beberapa Wilayah secara bersamaan).
- Untuk mode Konkurensi, pilih bagaimana konkurensi berperilaku selama eksekusi operasi.
 - Toleransi kegagalan yang ketat — Mengurangi tingkat konkurensi akun saat kegagalan terjadi, tetap berada dalam toleransi Kegagalan +1.
 - Toleransi kegagalan lunak - Mempertahankan tingkat konkurensi yang Anda tentukan (nilai akun bersamaan maksimum) terlepas dari kegagalan.

d. Pilih Berikutnya.

6. Pada halaman Specify Overrides, pilih kotak centang untuk parameter yang akan diganti, lalu pilih Override StackSetValue dari menu Edit override value.
7. Pada halaman Nilai StackSet parameter Override, buat perubahan, lalu pilih Simpan perubahan.

 Note

Untuk menyetel parameter yang diganti kembali menggunakan nilai yang ditentukan dalam StackSet, periksa semua parameter dan pilih Set to StackSet value dari menu Edit override value. Melakukannya akan menghapus semua nilai yang diganti setelah Anda memperbarui tumpukan.

8. Pada halaman Tinjau, tinjau pilihan Anda. Untuk membuat perubahan, pilih Edit pada bagian terkait.
9. Saat Anda siap untuk melanjutkan, pilih Kirim.

CloudFormation mulai memperbarui tumpukan Anda. Lihat kemajuan dan status tumpukan di halaman StackSet detail yang terbuka saat Anda memilih Kirim.

Ganti parameter pada tumpukan ()AWS CLI

Note

Saat bertindak sebagai administrator yang didelegasikan, Anda harus memasukkan `--call-as DELEGATED_ADMIN` dalam perintah.

Untuk mengganti parameter untuk tumpukan tertentu

1. Gunakan [update-stack-instances](#) AWS CLI perintah dan tentukan `--parameter-overrides` opsi.

[Izin yang dikelola sendiri] Untuk `--accounts` opsi ini, berikan akun IDs yang ingin Anda ganti nilai parameternya pada tumpukan.

```
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
```

[Izin yang dikelola layanan] Untuk `--deployment-targets` opsi ini, berikan ID root organisasi, OU IDs, atau AWS Organizations akun yang ingin Anda IDs ganti parameternya pada tumpukan. Dalam contoh ini, kami mengganti nilai parameter untuk tumpukan di semua akun di OU dengan ID. *ou-rcuk-1x5j1lwo*

Nilai parameter yang diganti hanya berlaku untuk akun yang saat ini berada di OU target dan anaknya. OUs Akun yang ditambahkan ke target OU dan anaknya OUs di masa depan akan menggunakan nilai StackSet default dan bukan nilai yang diganti.

```
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. Verifikasi bahwa nilai parameter Anda berhasil diganti pada tumpukan dengan menjalankan `describe-stack-set-operation` perintah untuk menampilkan status dan hasil operasi pembaruan

Anda. Untuk `--operation-id`, gunakan ID operasi yang dikembalikan oleh perintah `update-stack-instances` Anda.

```
aws cloudformation describe-stack-set-operation --operation-id operation_ID
```

Hapus tumpukan dari AWS CloudFormation StackSets

Anda dapat menghapus tumpukan dari StackSets menggunakan CloudFormation konsol atau. AWS CLI

Topik

- [Hapus tumpukan dari StackSet \(konsol\)](#)
- [Hapus tumpukan dari StackSet \(\)AWS CLI](#)

Note

Menghapus tumpukan dari unit organisasi tingkat atas (OU) menghapus OU tersebut sebagai target. StackSet

Hapus tumpukan dari StackSet (konsol)

Untuk menghapus tumpukan Anda

1. Masuk ke AWS Management Console dan buka AWS CloudFormation konsol di <https://console.aws.amazon.com/cloudformation>.
2. Pada bilah navigasi di bagian atas layar, pilih yang Wilayah AWS Anda buat StackSet di.
3. Dari panel navigasi, pilih StackSets. Pada StackSets halaman, pilih StackSet.
4. Dengan StackSet pilihan Anda, pilih Hapus tumpukan StackSet dari menu Tindakan.
5. Pada halaman Setel opsi penerapan, pertama-tama pilih akun dan Wilayah tempat Anda ingin menghapus tumpukan.
 - a. [Izin yang dikelola sendiri] Untuk Akun, pilih Menerapkan tumpukan di akun atau Menerapkan tumpukan di unit organisasi.

Jika Anda memilih Menyebarkan tumpukan di akun, tempel nomor akun target Anda di kotak teks Nomor akun, pisahkan beberapa angka dengan koma.

Jika Anda memilih Menerapkan tumpukan di unit organisasi, tempelkan ID OU target di kotak teks Nomor organisasi untuk menargetkan semua akun yang merupakan bagian dari organisasi yang ditentukan.

- b. [Izin yang dikelola layanan] Untuk unit Organisasi (OUs), tentukan target OU. IDs

 Important

CloudFormation akan menghapus tumpukan dari target yang ditentukan OUs dan anak OUs mereka.

Untuk jenis filter Akun, Anda dapat memperbaiki akun mana yang akan menghapus tumpukan dengan memilih salah satu opsi berikut dan memberikan nomor akun.

- Tidak ada (default) - Hapus tumpukan dari semua akun yang ditentukan OUs.
- Persimpangan — Hapus tumpukan hanya dari akun individu tertentu dalam yang dipilih. OUs
- Perbedaan — Hapus tumpukan dari semua akun di akun yang dipilih OUs kecuali untuk akun tertentu.
- Union — Hapus tumpukan dari akun individual tambahan yang ditentukan OUs ditambah.

- c. Untuk Tentukan wilayah, pilih Wilayah tempat Anda ingin menghapus tumpukan dalam akun target.

6. Untuk opsi Deployment, lakukan hal berikut:

- Untuk Akun bersamaan maksimum, tentukan berapa banyak akun yang diproses secara bersamaan.
- Untuk toleransi Kegagalan, tentukan jumlah maksimum kegagalan akun yang diizinkan per Wilayah. Operasi akan berhenti dan tidak akan melanjutkan ke Wilayah lain setelah batas ini tercapai.
- Untuk Pertahankan tumpukan, aktifkan opsi ini untuk menyimpan tumpukan dan sumber daya terkait saat menghapusnya dari tumpukan Anda. StackSet Sumber daya tetap dalam keadaan mereka saat ini tetapi tidak lagi menjadi bagian dari StackSet.

- Untuk konkurensi Wilayah, pilih cara memproses Wilayah: Berurutan (satu Wilayah pada satu waktu) atau Paralel (beberapa Wilayah secara bersamaan).
 - Untuk mode Konkurensi, pilih bagaimana konkurensi berperilaku selama eksekusi operasi.
 - Toleransi kegagalan yang ketat — Mengurangi tingkat konkurensi akun saat kegagalan terjadi, tetap berada dalam toleransi Kegagalan +1.
 - Toleransi kegagalan lunak - Mempertahankan tingkat konkurensi yang Anda tentukan (nilai akun bersamaan maksimum) terlepas dari kegagalan.
7. Pilih Berikutnya.
 8. Pada halaman Tinjau, tinjau pilihan Anda. Untuk membuat perubahan, pilih Edit pada bagian terkait.
 9. Ketika Anda siap untuk menghapus tumpukan dari Anda StackSet, pilih Kirim.

Setelah penghapusan tumpukan selesai, Anda dapat memverifikasi bahwa tumpukan telah dihapus dari halaman StackSet detail Anda StackSet , pada tab Stack instance.

Hapus tumpukan dari StackSet ()AWS CLI

Note

Saat bertindak sebagai administrator yang didelegasikan, Anda harus memasukkan `--call-as DELEGATED_ADMIN` dalam perintah.

Gunakan `delete-stack-instances` perintah dengan StackSet nama Anda.

Dalam contoh ini, kami menggunakan `--no-retain-stacks` opsi karena kami tidak mempertahankan tumpukan apa pun. Gunakan `--retain-stacks` alih-alih `--no-retain-stacks` jika Anda ingin menyimpan tumpukan dan sumber dayanya.

Untuk `--regions`, tentukan tumpukan yang ingin Wilayah AWS Anda hapus, misalnya, `us-west-2` dan `us-east-1`.

Tetapkan pemrosesan akun bersamaan dan preferensi lainnya menggunakan `--operation-preferences` opsi. Contoh-contoh ini menggunakan pengaturan berbasis hitungan. Perhatikan bahwa tidak `MaxConcurrentCount` boleh melebihi `FailureToleranceCount + 1`.

Untuk pengaturan berbasis persentase, gunakan `FailureTolerancePercentage` atau `MaxConcurrentPercentage` sebagai gantinya.

Untuk menghapus tumpukan (izin yang dikelola sendiri)

Untuk `--accounts` opsi, tentukan IDs akun untuk menghapus tumpukan dari.

```
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
```

Untuk menghapus tumpukan (izin yang dikelola layanan)

Untuk `--deployment-targets`, tentukan ID root organisasi atau unit organisasi (OU) IDs untuk menghapus tumpukan.

 Important

CloudFormation akan menghapus tumpukan dari target yang ditentukan OUs dan anak OUs mereka.

```
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
```

Untuk informasi selengkapnya, lihat [DeleteStackInstances](#) di dalam Referensi API AWS CloudFormation .

Secara opsional, setelah penghapusan tumpukan selesai, verifikasi bahwa tumpukan telah dihapus dari Anda StackSet dengan menjalankan `describe-stack-set-operation` perintah untuk menampilkan status dan hasil operasi tumpukan hapus. Untuk `--operation-id`, gunakan ID operasi yang dikembalikan oleh perintah `delete-stack-instances` Anda.

```
aws cloudformation describe-stack-set-operation --stack-set-name my-stackset \  
  --operation-id ddf16f54-ad62-4d9b-b0ab-3ed8e9example
```


Hapus AWS CloudFormation StackSets

Untuk menghapus StackSet, Anda harus terlebih dahulu menghapus semua tumpukan di StackSet. Untuk informasi tentang cara menghapus semua tumpukan, lihat [Hapus tumpukan dari StackSets](#).

Topik

- [Hapus StackSet \(konsol\)](#)
- [Hapus a StackSet \(AWS CLI\)](#)
- [Hapus peran layanan \(opsional\)](#)

Hapus StackSet (konsol)

Untuk menghapus StackSet

1. Masuk ke AWS Management Console dan buka AWS CloudFormation konsol di <https://console.aws.amazon.com/cloudformation>.
2. Pada bilah navigasi di bagian atas layar, pilih yang Wilayah AWS Anda buat StackSet di.
3. Pada StackSetshalaman, pilih StackSet.
4. Dengan yang StackSet dipilih, pilih Hapus StackSet dari menu Tindakan.
5. Ketika Anda diminta untuk mengonfirmasi bahwa Anda ingin menghapus StackSet, pilih Hapus StackSet.

Hapus a StackSet (AWS CLI)

Note

Saat bertindak sebagai administrator yang didelegasikan, Anda harus memasukkan `--call-as DELEGATED_ADMIN` dalam perintah.

Untuk menghapus StackSet

1. Gunakan perintah berikut delete-stack-set. Saat Anda diminta untuk mengonfirmasi, ketik **y**, lalu tekan Masuk.

```
aws cloudformation delete-stack-set --stack-set-name my-stackset
```

2. Verifikasi StackSet bahwa telah dihapus dengan menjalankan list-stack-sets perintah. Hasil list-stack-sets perintah harus menunjukkan tumpukan Anda dengan status DELETED.

```
aws cloudformation list-stack-sets
```

Hapus peran layanan (opsional)

Jika Anda tidak lagi memerlukan peran layanan IAM yang CloudFormation diperlukan untuk melakukan StackSet operasi, sebaiknya Anda menghapus peran tersebut.

Untuk dikelola sendiri StackSets, peran yang Anda buat. Untuk informasi lebih lanjut tentang peran ini, lihat [Memberikan izin yang dikelola sendiri](#).

Untuk pengelolaan layanan StackSets, peran yang dibuat secara otomatis StackSets memiliki akhiran CloudFormationStackSetsOrgAdmin di akun manajemen organisasi, dan CloudFormationStackSetsOrgMember di setiap akun target. Untuk informasi selengkapnya, lihat [Peran terkait layanan](#).

Untuk menghapus peran layanan (konsol)

1. Masuk ke AWS Management Console dan buka konsol IAM di <https://console.aws.amazon.com/iam/>.
2. Dari panel navigasi, pilih Peran, lalu isi kotak centang di samping peran yang ingin Anda hapus.
3. Untuk menu Tindakan peran di bagian atas halaman, pilih Hapus peran.
4. Di kotak dialog konfirmasi, pilih Ya, Hapus. Jika Anda yakin, Anda dapat melanjutkan penghapusan meskipun data terakhir mengakses layanan masih memuat.

Untuk menghapus peran layanan (AWS CLI)

- Gunakan perintah berikut delete-role. Saat Anda diminta untuk mengonfirmasi, ketik **y**, lalu tekan Enter.

```
aws iam delete-role --role-name role name
```

Untuk informasi selengkapnya tentang menghapus peran, lihat [Menghapus peran atau profil instance](#) di Panduan Pengguna IAM.

Mencegah StackSets penerapan yang gagal menggunakan gerbang akun target

Gerbang akun adalah fitur opsional yang membantu Anda memverifikasi bahwa akun target memenuhi persyaratan tertentu sebelum CloudFormation memulai StackSets operasi di akun tersebut. Verifikasi ini dilakukan melalui AWS Lambda fungsi yang bertindak sebagai pemeriksaan prasyarat.

Contoh umum dari gerbang akun adalah memverifikasi bahwa tidak ada CloudWatch alarm yang aktif atau tidak terselesaikan pada akun target. CloudFormation memanggil fungsi Lambda setiap kali Anda memulai operasi tumpukan di akun target, dan hanya berlanjut jika fungsi mengembalikan SUCCEEDED kode. Jika fungsi Lambda mengembalikan statusFAILED, CloudFormation tidak melanjutkan operasi yang Anda minta. Jika Anda tidak memiliki fungsi Lambda gerbang akun yang dikonfigurasi CloudFormation, lewati pemeriksaan, dan lanjutkan operasi Anda.

Jika akun target Anda menggagalkan pemeriksaan gerbang akun, operasi yang digagalkan akan dihitung terhadap jumlah toleransi kegagalan yang ditentukan atau persentase tumpukan. Untuk informasi lebih lanjut tentang toleransi kegagalan, lihat [StackSet opsi operasi](#).

Account gating hanya tersedia untuk StackSets operasi. Fitur ini tidak tersedia untuk CloudFormation operasi lain di luar StackSets.

Persyaratan

Persyaratan berikut harus dipenuhi untuk account gating:

- Fungsi Lambda Anda harus diberi nama `AWSCloudFormationStackSetAccountGate` untuk menggunakan fitur ini.
- `AWSCloudFormationStackSetExecutionRole` harus memiliki izin untuk menjalankan fungsi Lambda Anda. Tanpa izin ini, CloudFormation akan melewati pemeriksaan gerbang akun dan melanjutkan dengan operasi tumpukan.
- `InvokeFunction` izin Lambda harus ditambahkan ke akun target agar gerbang akun berfungsi. Kebijakan kepercayaan akun target harus memiliki hubungan kepercayaan dengan akun administrator. Berikut ini adalah contoh pernyataan kebijakan yang memberikan izin `InvokeFunction` Lambda.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "lambda:InvokeFunction"
      ],
      "Resource": "*"
    }
  ]
}
```

CloudFormation template untuk membuat fungsi Lambda

Gunakan contoh template berikut untuk membuat fungsi LambdaAWS CloudFormation StackSet AccountGate. Untuk membuat tumpukan baru menggunakan salah satu templat ini, lihat [Buat tumpukan dari CloudFormation konsol](#).

Lokasi templat	Deskripsi
https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/AccountGateSucceeded.yml	Membuat tumpukan yang mengimplementasikan fungsi gerbang akun Lambda yang akan mengembalikan status. SUCCEEDED
https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/AccountGateFailed.yml	Membuat tumpukan yang mengimplementasikan fungsi gerbang akun Lambda yang akan mengembalikan status. FAILED

Pilih Mode Konkurensi untuk AWS CloudFormation StackSets

Mode Konkurensi adalah parameter [StackSetOperationPreferences](#) yang memungkinkan Anda memilih bagaimana tingkat konkurensi berperilaku selama operasi. StackSet Anda dapat memilih di antara mode berikut:

- Toleransi Kegagalan Ketat: Opsi ini secara dinamis menurunkan tingkat konkurensi untuk memastikan jumlah akun yang gagal tidak pernah melebihi nilai toleransi Kegagalan +1.

Konkurensi aktual awal diatur ke nilai yang lebih rendah dari nilai akun bersamaan Maksimum, atau nilai toleransi Kegagalan +1. Konkurensi aktual kemudian dikurangi secara proporsional dengan jumlah kegagalan. Ini adalah perilaku default.

- **Soft Failure Tolerance:** Opsi ini memisahkan toleransi kegagalan dari konkurensi yang sebenarnya. Ini memungkinkan StackSet operasi berjalan pada tingkat konkurensi yang ditetapkan oleh nilai akun bersamaan maksimum, terlepas dari jumlah kegagalan.

Toleransi Kegagalan Ketat menurunkan kecepatan penerapan karena kegagalan StackSet operasi terjadi karena konkurensi menurun untuk setiap kegagalan. Soft Failure Tolerance memprioritaskan kecepatan penerapan sambil tetap memanfaatkan CloudFormation kemampuan keselamatan. Ini memungkinkan Anda untuk meninjau dan mengatasi kegagalan StackSet operasi untuk masalah umum seperti yang terkait dengan sumber daya yang ada, kuota layanan, dan izin.

Untuk informasi selengkapnya tentang kegagalan operasi StackSets tumpukan, lihat [Alasan umum untuk kegagalan operasi tumpukan](#).

Untuk informasi selengkapnya tentang Akun bersamaan maksimum dan Toleransi kegagalan, lihat [StackSet opsi operasi](#).

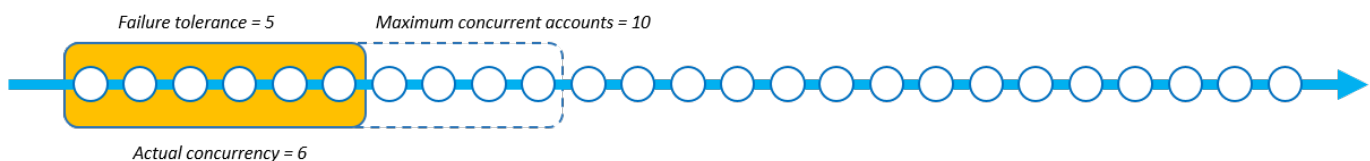
Cara kerja setiap Mode Konkurensi

Gambar di bawah ini memberikan representasi visual tentang bagaimana setiap Mode Konkurensi bekerja selama StackSet operasi. String node mewakili penyebaran ke tunggal Wilayah AWS dan setiap node adalah target Akun AWS.

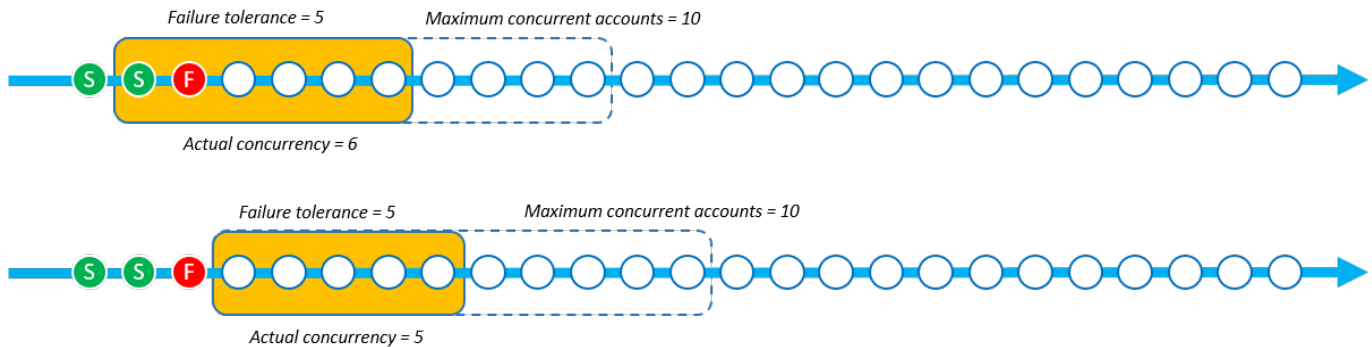
Toleransi Kegagalan yang Ketat

Ketika StackSet operasi menggunakan Toleransi Kegagalan Ketat memiliki nilai toleransi Kegagalan ditetapkan ke 5 dan nilai akun bersamaan Maksimum ditetapkan ke 10, konkurensi sebenarnya adalah 6. Konkurensi aktual adalah 6 karena ini nilai toleransi Kegagalan 5 +1 lebih rendah dari nilai akun bersamaan Maksimum.

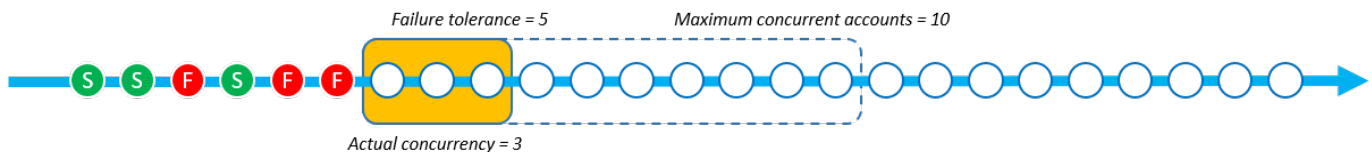
Gambar berikut menunjukkan dampak nilai toleransi Kegagalan terhadap nilai akun bersamaan Maksimum, dan dampaknya terhadap konkurensi aktual operasi: StackSet



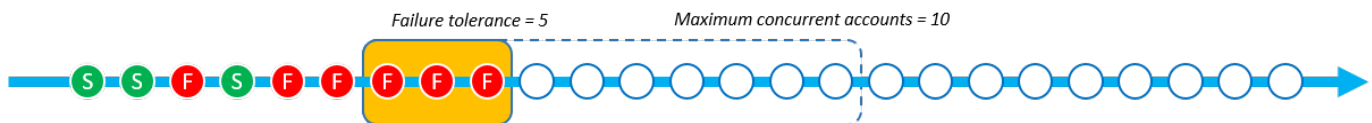
Ketika penerapan dimulai dan ada instance tumpukan yang gagal, maka konkurensi sebenarnya berkurang untuk memberikan pengalaman penerapan yang aman. Konkurensi aktual berkurang dari 6 menjadi 5 ketika StackSets gagal menerapkan 1 instance tumpukan.



Mode Toleransi Kegagalan Ketat mengurangi konkurensi aktual secara proporsional dengan jumlah instance tumpukan yang gagal. Dalam contoh berikut, konkurensi aktual berkurang dari 5 menjadi 3 ketika StackSets gagal menerapkan 2 instance tumpukan lagi, sehingga total instance tumpukan yang gagal menjadi 3.



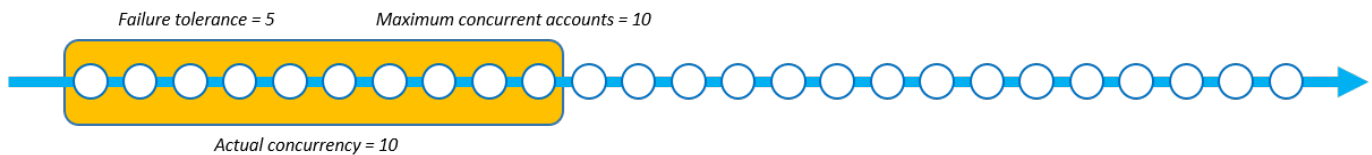
StackSets gagal StackSet operasi ketika jumlah instance tumpukan gagal sama dengan nilai yang ditentukan dari toleransi Kegagalan +1. Dalam contoh berikut, StackSets gagal operasi ketika ada 6 instance tumpukan gagal dan nilai toleransi kegagalan adalah 5.



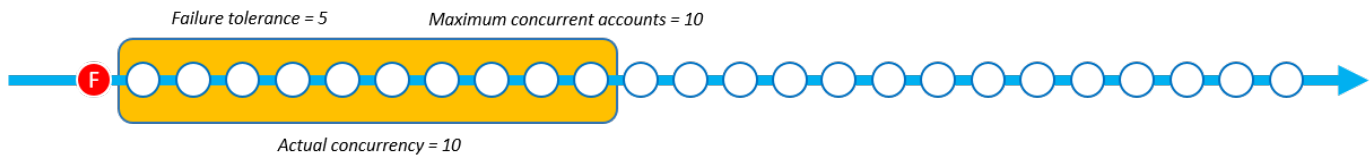
Dalam contoh ini, CloudFormation menerapkan 9 instance tumpukan (3 berhasil dan 6 gagal) sebelum menghentikan operasi. StackSet

Toleransi Kegagalan Lunak

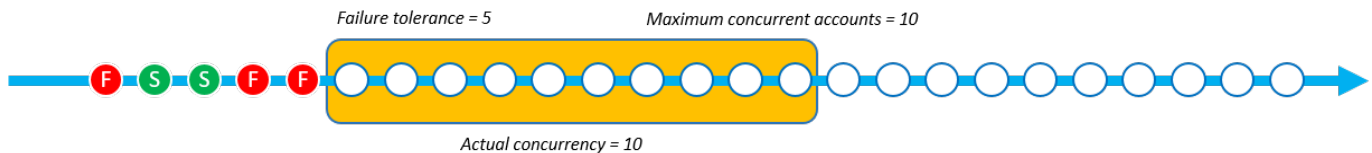
Ketika StackSet operasi menggunakan Soft Failure Tolerance memiliki nilai toleransi Kegagalan ditetapkan ke 5 dan nilai akun bersamaan Maksimum ditetapkan ke 10, konkurensi sebenarnya adalah 10.



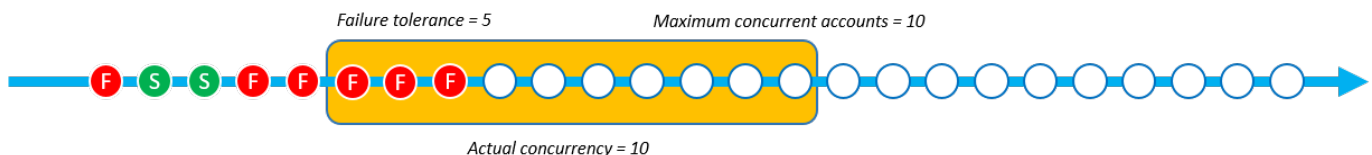
Saat penerapan dimulai dan ada instance tumpukan yang gagal, konkurensi sebenarnya tidak berubah. Dalam contoh berikut, 1 operasi tumpukan gagal, tetapi konkurensi sebenarnya tetap pada 10.



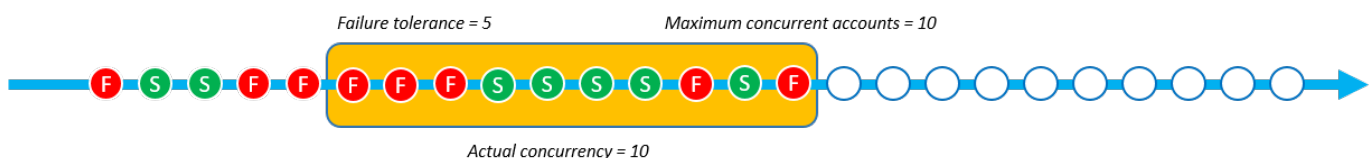
Konkurensi aktual tetap pada 10 bahkan setelah 2 kegagalan instance tumpukan lagi.



StackSets gagal StackSet operasi ketika instance tumpukan gagal melebihi nilai toleransi Kegagalan. Dalam contoh berikut, StackSets gagal operasi ketika ada 6 instance tumpukan gagal dan jumlah toleransi kegagalan adalah 5. Namun, operasi tidak akan berakhir sampai operasi yang tersisa dalam antrian konkurensi selesai.



StackSets terus menerapkan instance tumpukan yang sudah ada dalam antrian konkurensi. Ini berarti bahwa jumlah instance tumpukan gagal bisa lebih tinggi daripada toleransi Kegagalan. Dalam contoh berikut, ada 8 instance tumpukan gagal karena antrian konkurensi masih memiliki 7 operasi yang tersisa untuk dilakukan, meskipun StackSet operasi telah mencapai toleransi Kegagalan 5.



Dalam contoh ini, StackSets menerapkan 15 instance tumpukan (7 berhasil dan 8 gagal) sebelum menghentikan operasi tumpukan.

Memilih antara toleransi kegagalan yang ketat dan toleransi kegagalan lunak berdasarkan kecepatan penerapan

Memilih antara Toleransi kegagalan yang ketat dan mode toleransi kegagalan lunak bergantung pada kecepatan StackSet penerapan yang diinginkan dan jumlah kegagalan penerapan yang diizinkan.

Tabel berikut menunjukkan bagaimana setiap mode konkurensi menangani StackSet operasi yang gagal saat mencoba menerapkan 1000 instance tumpukan total. Dalam setiap skenario, nilai toleransi Kegagalan diatur ke 100 instance tumpukan dan nilai akun bersamaan Maksimum diatur ke 250 instance tumpukan.

Meskipun StackSets sebenarnya mengantri akun sebagai jendela geser (lihat [Cara kerja setiap Mode Konkurensi](#)), contoh ini menunjukkan operasi dalam batch untuk menunjukkan kecepatan setiap mode.

Toleransi kegagalan yang ketat

Contoh menggunakan mode toleransi kegagalan ketat ini menurunkan konkurensi aktual relatif terhadap jumlah kegagalan yang terjadi di setiap batch sebelumnya. Setiap batch memiliki 20 instance gagal, yang kemudian menurunkan konkurensi aktual dari batch berikut sebesar 20 hingga StackSet operasi mencapai nilai toleransi Kegagalan 100.

Dalam tabel berikut, konkurensi aktual awal dari batch pertama adalah 101 instance tumpukan. Konkurensi sebenarnya adalah 101 karena nilai yang lebih rendah dari akun bersamaan Maksimum (250) dan toleransi Kegagalan (100) +1. Setiap batch berisi 20 penerapan instans tumpukan yang gagal, yang kemudian menurunkan konkurensi aktual dari setiap batch berikut sebanyak 20 instance tumpukan.

Toleransi kegagalan yang ketat	Batch 1	Batch 2	Batch 3	Batch 4	Batch 5	Batch 6
Hitungan konkurensi aktual	101	81	61	41	21	-

Toleransi kegagalan yang ketat	Batch 1	Batch 2	Batch 3	Batch 4	Batch 5	Batch 6
Hitungan instans gagal	20	20	20	20	20	-
Jumlah instans tumpukan yang berhasil	81	61	41	21	1	-

Operasi yang menggunakan toleransi kegagalan yang ketat menyelesaikan 305 penerapan instans tumpukan dalam 5 batch pada saat StackSet operasi mencapai toleransi Kegagalan 100 instance tumpukan. StackSet Operasi berhasil menyebarkan 205 instance tumpukan sebelum gagal.

Toleransi kegagalan lunak

Contoh menggunakan mode toleransi kegagalan lunak ini mempertahankan jumlah konkurensi aktual yang sama yang ditentukan oleh nilai akun bersamaan Maksimum dari 250 instance tumpukan, terlepas dari jumlah instans yang gagal. StackSet Operasi menyimpan konkurensi aktual yang sama hingga mencapai nilai toleransi Kegagalan 100 instance.

Dalam tabel berikut, konkurensi aktual awal dari batch pertama adalah 250 instance tumpukan. Konkurensi aktual adalah 250 karena nilai akun bersamaan Maksimum diatur ke 250 dan mode toleransi kegagalan lunak memungkinkan StackSets untuk menggunakan nilai ini sebagai konkurensi aktual, terlepas dari jumlah kegagalan. Meskipun ada 50 kegagalan di masing-masing batch untuk contoh ini, konkurensi sebenarnya tetap tidak terpengaruh.

Toleransi kegagalan lunak	Batch 1	Batch 2	Batch 3	Batch 4	Batch 5	Batch 6
Hitungan konkurensi aktual	250	250	-	-	-	-
Hitungan instans gagal	50	50	-	-	-	-
Jumlah instans tumpukan yang berhasil	200	200	-	-	-	-

Menggunakan nilai akun bersamaan Maksimum dan nilai toleransi Kegagalan yang sama, operasi menggunakan mode toleransi kegagalan lunak menyelesaikan 500 penerapan instans tumpukan dalam 2 batch. StackSet Operasi berhasil menyebarkan 400 instance tumpukan sebelum gagal.

Memilih Mode Konkurensi (konsol)

Saat membuat atau memperbarui StackSet, pada halaman Setel opsi penerapan, untuk mode Konkurensi, pilih Toleransi kegagalan yang ketat atau Toleransi kegagalan lunak.

Memilih Mode Konkurensi Anda (AWS CLI)

Anda dapat menggunakan `ConcurrencyMode` parameter dengan StackSets perintah berikut:

- [create-stack-instances](#)
- [delete-stack-instances](#)
- [detect-stack-set-drift](#)
- [import-stacks-to-stack-set](#)
- [update-stack-instances](#)

- [update-stack-set](#)

Perintah ini memiliki parameter yang ada `--operation-preferences` yang disebut yang dapat menggunakan `ConcurrencyMode` pengaturan. `ConcurrencyMode` dapat diatur ke salah satu nilai berikut:

- `STRICT_FAILURE_TOLERANCE`
- `SOFT_FAILURE_TOLERANCE`

Contoh berikut membuat instance stack menggunakan `STRICT_FAILURE_TOLERANCE` `ConcurrencyMode`, dengan `FailureToleranceCount` set ke 10 dan `MaxConcurrentCount` set ke 5.

```
aws cloudformation create-stack-instances \  
  --stack-set-name example-stackset \  
  --accounts 123456789012 \  
  --regions eu-west-1 \  
  --operation-preferences  
    ConcurrencyMode=STRICT_FAILURE_TOLERANCE,FailureToleranceCount=10,MaxConcurrentCount=5
```

Note

Untuk prosedur rinci untuk membuat dan memperbarui StackSet, lihat topik berikut:

- [Buat StackSets \(izin yang dikelola sendiri\)](#)
- [Buat StackSets \(izin yang dikelola layanan\)](#)
- [Perbarui StackSets](#)

Melakukan deteksi drift pada CloudFormation StackSets

Bahkan saat Anda mengelola tumpukan dan sumber daya yang dikandungnya CloudFormation, pengguna dapat mengubah sumber daya tersebut di luar. CloudFormation Pengguna dapat mengedit sumber daya secara langsung dengan menggunakan layanan dasar yang membuat sumber daya. Dengan melakukan deteksi drift pada a StackSet, Anda dapat menentukan apakah salah satu instance tumpukan milik yang StackSet berbeda, atau telah melayang, dari konfigurasi yang diharapkan.

Topik

- [Bagaimana CloudFormation melakukan deteksi drift pada StackSet](#)
- [Mendeteksi penyimpangan pada StackSet \(konsol\)](#)
- [Mendeteksi penyimpangan pada StackSet \(AWS CLI\)](#)
- [Menghentikan deteksi drift pada StackSet](#)

Bagaimana CloudFormation melakukan deteksi drift pada StackSet

Saat CloudFormation melakukan deteksi drift pada a StackSet, ia melakukan deteksi drift pada tumpukan yang terkait dengan setiap instance tumpukan di file. StackSet Untuk melakukan ini, CloudFormation membandingkan keadaan terkini setiap sumber daya dalam tumpukan dengan perkiraan keadaan sumber daya tersebut, sebagaimana ditetapkan dalam templat tumpukan dan parameter input tertentu. Jika keadaan sumber daya terkini berbeda dari keadaan yang diharapkan, sumber daya tersebut dianggap telah menyimpang. Jika satu atau beberapa sumber daya dalam tumpukan telah menyimpan, maka tumpukan itu sendiri dianggap telah menyimpan, dan instans tumpukan yang terkait dengan tumpukan dianggap telah menyimpang juga. Jika satu atau lebih instance tumpukan dalam StackSet telah melayang, StackSet itu sendiri dianggap telah hanyut.

Deteksi drift mengidentifikasi perubahan yang tidak dikelola; yaitu, perubahan yang dilakukan pada tumpukan di luar. CloudFormation Perubahan yang dilakukan melalui CloudFormation tumpukan secara langsung, bukan pada StackSet level, tidak dianggap hanyut. Misalnya, Anda memiliki tumpukan yang terkait dengan instance tumpukan a StackSet. Jika Anda menggunakan CloudFormation untuk memperbarui tumpukan itu untuk menggunakan template yang berbeda, itu tidak dianggap drift, meskipun tumpukan itu sekarang memiliki template yang berbeda dari tumpukan lain milik. StackSet Hal ini karena tumpukan masih menyesuaikan templat yang diperkirakan dan konfigurasi parameter dalam CloudFormation.

Untuk informasi rinci tentang cara CloudFormation melakukan deteksi drift pada tumpukan, lihat [Mendeteksi perubahan konfigurasi yang tidak terkelola pada tumpukan dan sumber daya dengan deteksi drift](#).

Karena CloudFormation melakukan deteksi drift pada setiap tumpukan secara individual, diperlukan nilai parameter yang diganti saat menentukan apakah tumpukan telah melayang. Untuk informasi lebih lanjut tentang menimpa parameter templat dalam instans tumpukan, lihat [Ganti nilai parameter pada tumpukan di dalam CloudFormation StackSet](#).

Jika Anda melakukan deteksi drift [secara langsung pada tumpukan yang](#) terkait dengan instance tumpukan, hasil drift tersebut tidak tersedia dari halaman StackSetskonsol.

Mendeteksi penyimpangan pada StackSet (konsol)

Untuk mendeteksi penyimpangan pada a StackSet

1. Buka AWS CloudFormation konsol di <https://console.aws.amazon.com/cloudformation>.
2. Pada StackSetshalaman, pilih StackSet di mana Anda ingin melakukan deteksi drift.
3. Dari menu Tindakan, pilih Deteksi penyimpangan.

CloudFormation menampilkan bilah informasi yang menyatakan bahwa deteksi drift telah dimulai untuk yang dipilih. StackSet

4. Opsional: Untuk memantau kemajuan operasi deteksi penyimpangan:
 - a. Pilih StackSet nama untuk menampilkan halaman detail Stackset.
 - b. Pilih tab Operasi, pilih operasi deteksi penyimpangan, lalu pilih Tampilkan detail penyimpangan.

CloudFormation menampilkan kotak dialog Detail operasi.

5. Tunggu hingga CloudFormation menyelesaikan operasi deteksi drift. Saat operasi deteksi drift selesai, CloudFormation perbarui status Drift dan waktu pemeriksaan drift terakhir untuk Anda. StackSet Bidang ini tercantum di tab Ikhtisar halaman StackSet detail untuk yang dipilih StackSet.

Operasi deteksi drift mungkin memakan waktu, tergantung pada jumlah instance tumpukan yang disertakan dalam StackSet, dan jumlah sumber daya yang disertakan dalam file. StackSet Anda hanya dapat menjalankan operasi deteksi drift tunggal pada yang diberikan StackSet pada satu waktu. CloudFormation melanjutkan operasi deteksi drift bahkan setelah Anda mengabaikan bilah informasi.

6. Untuk meninjau hasil deteksi drift untuk instance tumpukan di a StackSet, pilih tab Stack instance.

Kolom Nama tumpukan mencantumkan nama tumpukan yang terkait dengan setiap instans tumpukan, dan kolom Status drift mencantumkan status penyimpangan tumpukan tersebut. Tumpukan dianggap telah menyimpang jika satu atau beberapa sumber dayanya telah menyimpang.

7. Untuk meninjau hasil deteksi drift untuk tumpukan yang terkait dengan instance tumpukan tertentu:
 - a. Pilih tab Operasi.
 - b. Pilih operasi drift yang ingin Anda lihat hasil deteksi drift. Panel split akan menampilkan status instance stack dan alasan untuk operasi yang dipilih. Untuk operasi drift, kolom alasan status menunjukkan status drift dari instance tumpukan.
 - c. Pilih instance tumpukan yang ingin Anda lihat detail drift, dan pilih Lihat drift sumber daya. Dalam tabel status drift Resource pada halaman Resource Drifts, setiap sumber daya tumpukan terdaftar dengan status driftnya dan deteksi drift terakhir kali dimulai pada sumber daya. ID logis dan ID fisik setiap sumber daya ditampilkan untuk membantu Anda mengidentifikasinya.
8. Anda dapat mengurutkan sumber daya berdasarkan status penyimpangannya menggunakan kolom Status Penyimpangan.

Untuk melihat detail pada sumber daya yang dimodifikasi:

- Dengan sumber daya yang dipilih, pilih Lihat detail drift.

CloudFormation menampilkan halaman detail drift untuk sumber daya tertentu. Halaman ini mencantumkan perbedaan sumber daya. Ini juga mencantumkan nilai properti yang diharapkan dan saat ini dari sumber daya.

 Note

Jika tumpukan milik Wilayah dan akun yang berbeda dari yang saat ini Anda masuki, tombol Deteksi drift akan dinonaktifkan dan Anda tidak akan dapat melihat detailnya.

Mendeteksi penyimpangan pada StackSet (AWS CLI)

Untuk mendeteksi penyimpangan pada seluruh tumpukan menggunakan AWS CLI, gunakan prosedur berikut:

Untuk mendeteksi penyimpangan pada a StackSet

1. Gunakan [detect-stack-set-drift](#) perintah untuk mendeteksi penyimpangan pada keseluruhan StackSet dan instance tumpukan yang terkait.

Contoh berikut memulai deteksi drift pada file. StackSet `stack-set-drift-example`

```
aws cloudformation detect-stack-set-drift \  
  --stack-set-name stack-set-drift-example
```

Output:

```
{  
  "OperationId": "c36e44aa-3a83-411a-b503-cb611example"  
}
```

2. Karena operasi deteksi StackSet drift dapat menjadi operasi yang berjalan lama, gunakan [describe-stack-set-operation](#) perintah untuk memantau status operasi drift. Perintah ini mengambil ID StackSet operasi yang dikembalikan oleh `detect-stack-set-drift` perintah.

Contoh berikut menggunakan ID operasi dari contoh sebelumnya untuk mengembalikan informasi tentang operasi deteksi StackSet drift. Dalam contoh ini, operasi masih berjalan. Dari tujuh instance tumpukan yang terkait dengan ini StackSet, satu instance tumpukan telah ditemukan telah hanyut, dua instance sinkron, dan deteksi drift untuk empat instance tumpukan yang tersisa masih dalam proses. Karena satu contoh telah melayang, status drift dari StackSet dirinya sendiri sekarang. DRIFTED

```
aws cloudformation describe-stack-set-operation \  
  --stack-set-name stack-set-drift-example \  
  --operation-id c36e44aa-3a83-411a-b503-cb611example
```

Output:

```
{  
  "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"
}
}

```

Melakukan perintah yang sama nanti, contoh ini menunjukkan informasi akan kembali setelah operasi deteksi penyimpangan selesai. Dua dari tujuh instance tumpukan total yang terkait dengan ini StackSet telah melayang, menjadikan status drift itu sendiri sebagai. StackSet DRIFTED

```

aws cloudformation describe-stack-set-operation \
  --stack-set-name stack-set-drift-example \
  --operation-id c36e44aa-3a83-411a-b503-cb611example

```

Output:

```

{
  "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. Ketika operasi deteksi StackSet drift selesai, gunakan `describe-stack-set`, `list-stack-instances`, `describe-stack-instance`, dan `list-stack-instance-resource-drifts` perintah untuk meninjau hasilnya.

Perintah [describe-stack-set](#) mencakup informasi penyimpangan terperinci yang sama yang dikembalikan oleh perintah `describe-stack-set-operation`.

```

aws cloudformation describe-stack-set \
    --stack-set-name stack-set-drift-example

```

Output:

```

{
  "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"
}
}

```

Anda dapat menggunakan [list-stack-instances](#) perintah untuk mengembalikan informasi ringkasan tentang instance tumpukan yang terkait dengan a StackSet, termasuk status drift dari setiap instance tumpukan.

Dalam contoh ini, mengeksekusi list-stack-instances pada contoh StackSet dengan filter status drift disetel untuk DRIFTED memungkinkan Anda mengidentifikasi dua instance tumpukan mana yang memiliki status drift. DRIFTED

```

aws cloudformation list-stack-instances \
  --stack-set-name stack-set-drift-example \
  --filters Name=DRIFT_STATUS,Values=DRIFTED

```

Output:

```

{
  "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]
]
}

```

Perintah [describe-stack-instance](#) juga mengembalikan informasi ini, tetapi untuk instans tumpukan tunggal, seperti dalam contoh di bawah ini.

```

aws cloudformation describe-stack-instance \
  --stack-set-name stack-set-drift-example \
  --stack-instance-account 012345678910 --stack-instance-region us-east-1

```

Output:

```

{
  "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. Setelah Anda mengidentifikasi instance tumpukan mana yang telah hanyut, Anda dapat menggunakan informasi tentang instance tumpukan yang dikembalikan oleh `describe-stack-instance` perintah `list-stack-instances` or untuk menjalankan perintah. [list-stack-instance-resource-drifts](#) Perintah ini mengembalikan informasi rinci tentang sumber daya dalam tumpukan yang telah hanyut untuk operasi drift tertentu.

Contoh berikut menggunakan `--stack-instance-resource-drift-statuses` parameter untuk meminta informasi stack drift untuk sumber daya yang telah dimodifikasi atau dihapus dalam contoh operasi drift sebelumnya. Permintaan mengembalikan informasi pada satu sumber daya yang telah dimodifikasi, termasuk detail tentang dua propertinya dan nilainya yang diubah. Tidak ada sumber daya yang telah dihapus.

```

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

```

Output:

```

{
  "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"
}
]
```

Menghentikan deteksi drift pada StackSet

Karena deteksi drift pada a StackSet bisa menjadi operasi yang berjalan lama, mungkin ada contoh ketika Anda ingin menghentikan operasi deteksi drift yang saat ini berjalan pada file. StackSet

Untuk menghentikan deteksi drift pada StackSet (konsol)

1. Buka AWS CloudFormation konsol di <https://console.aws.amazon.com/cloudformation>.
2. Pada StackSetshalaman, pilih nama StackSet.

CloudFormation menampilkan halaman StackSets detail untuk yang dipilih StackSet.

3. Pada halaman StackSets detail, pilih tab Operasi, lalu pilih operasi deteksi drift.
4. Pilih Hentikan operasi.

Untuk menghentikan deteksi drift pada StackSet (AWS CLI)

- Gunakan perintah [stop-stack-set-operation](#). Anda harus memberikan StackSet nama dan ID operasi StackSet operasi deteksi drift.

```
aws cloudformation stop-stack-set-operation \
  --stack-set-name stack-set-drift-example \
  --operation-id 624af370-311a-11e8-b6b7-500cexample
```

Impor tumpukan ke AWS CloudFormation StackSets

Operasi impor tumpukan dapat mengimpor tumpukan yang ada ke yang baru atau yang sudah ada StackSets, sehingga Anda dapat memigrasikan tumpukan yang ada ke StackSet dalam satu operasi.

Untuk dikelola sendiri StackSets, operasi impor dapat mengimpor tumpukan di akun administrator atau di akun target yang berbeda dan. Wilayah AWS Untuk pengelolaan layanan StackSets, operasi impor dapat mengimpor tumpukan apa pun yang AWS Organizations sama dengan akun manajemen.

Berikut ini adalah pertimbangan dan batasan saat mengimpor tumpukan ke: StackSets

- Operasi impor dapat mengimpor hingga 10 tumpukan menggunakan tumpukan sebaris IDs atau hingga 200 tumpukan menggunakan objek Amazon S3.
- Properti NoEcho tidak didukung. Tumpukan yang berisi NoEcho tidak akan diimpor ke baru StackSets melalui StackSet impor.
- Tumpukan hanya bisa menjadi milik satu StackSet.
- Anda dapat menerapkan tag tumpukan ke tag StackSet dengan menentukan tag secara eksplisit sebagai parameter dalam operasi impor tumpukan.
- Penimpaan parameter khusus tumpukan tidak terpengaruh selama operasi impor.
- StackSets Kuota dan instance tumpukan berlaku saat mengimpor tumpukan. Untuk informasi lebih lanjut tentang kuota, lihat [Memahami CloudFormation kuota](#).

Topik

- [Impor tumpukan yang dikelola sendiri untuk AWS CloudFormation StackSets](#)
- [Impor tumpukan yang dikelola layanan untuk AWS CloudFormation StackSets](#)
- [Kembalikan impor tumpukan ke AWS CloudFormation StackSets](#)

Impor tumpukan yang dikelola sendiri untuk AWS CloudFormation StackSets

Operasi impor CloudFormation tumpukan dapat mengimpor tumpukan yang ada ke yang baru atau yang sudah ada StackSets, sehingga Anda dapat memigrasikan tumpukan yang ada ke StackSet dalam satu operasi. Dengan menggunakan impor tumpukan, Anda menghindari waktu henti dan pemadaman tanpa menghapus dan membuat ulang sumber daya tersebut. Setelah tumpukan

diimpor ke dalam StackSet, tumpukan asli akan menjadi instance tumpukan dari kumpulan tumpukan yang ditentukan.

Pertimbangan untuk impor tumpukan yang dikelola sendiri

- Operasi impor tumpukan memerlukan akun administrator tempat Anda membuat StackSet dan akun target yang berisi tumpukan.
- Akun target harus memiliki izin untuk menggunakan `GetTemplate` operasi dengan input stack ID atau ARN. Karena itu, akun administrator Anda harus diberikan `AWSCloudFormationStackSetAdministrationRole` atau `AWSCloudFormationStackSetsExecutionRole` izin.

Topik

- [Impor tumpukan yang ada ke StackSet \(konsol\) baru](#)
- [Impor tumpukan yang ada ke StackSet \(konsol\) yang ada](#)
- [Impor tumpukan ke StackSet \(AWS CLI\)](#)

Impor tumpukan yang ada ke StackSet (konsol) baru

Sebelum Anda mulai, identifikasi tumpukan yang ingin Anda impor.

1. Masuk ke AWS Management Console dan buka AWS CloudFormation konsol di <https://console.aws.amazon.com/cloudformation>.
2. Dari panel navigasi, pilih StackSets.
3. Di bagian atas StackSetshalaman, pilih Buat StackSet.
4. Pada halaman Pilih templat, tentukan templat dengan salah satu opsi berikut dan pilih Berikutnya.
 - Pilih URL Amazon S3 dan tentukan URL untuk template Anda di kotak teks.
 - Pilih Unggah file templat dan telusuri templat Anda.
 - Pilih Dari ID tumpukan dan masukkan ID tumpukan Anda.
5. Pada halaman Tentukan StackSet detail, masukkan nama yang ingin StackSet Anda buat dan pilih Berikutnya.

(Opsional) Masukkan deskripsi dari StackSet.

6. Pada halaman Konfigurasi StackSet opsi, tinjau pilihan Anda dan pilih Berikutnya.
7. Pada halaman Setel opsi penerapan, pilih Impor tumpukan ke set tumpukan.

8. Masukkan ID tumpukan tumpukan yang ingin Anda impor di bidang Stacks to import. Misalnya, *arn:aws:cloudformation:us-east-1:123456789012:stack/StackToImport/f449b250-b969-11e0-a185-5081d0136786*.

(Opsional) Pilih Tambahkan ID tumpukan lain dan masukkan ID tumpukan tumpukan lain yang ingin Anda impor. Anda dapat menambahkan hingga 10 tumpukan per operasi impor tumpukan.

9. Tinjau opsi penerapan Anda dan pilih Berikutnya.

10. Pada halaman Ulasan, tinjau pilihan Anda StackSet dan properti Anda. Ketika Anda siap untuk mengimpor tumpukan Anda ke dalam Anda StackSet, pilih Kirim.

Hasil: Tumpukan yang diimpor sekarang merupakan instance tumpukan yang ditentukan StackSet. Untuk mempelajari lebih lanjut tentang status impor tumpukan, lihat [StackSets kode status](#).

Impor tumpukan yang ada ke StackSet (konsol) yang ada

Sebelum Anda mulai, identifikasi tumpukan yang ingin Anda impor.

1. Masuk ke AWS Management Console dan buka AWS CloudFormation konsol di <https://console.aws.amazon.com/cloudformation>.
2. Dari panel navigasi, pilih StackSets.
3. Pada StackSetshalaman, pilih tumpukan StackSet yang ingin Anda impor.
4. Dengan yang StackSet dipilih, pilih Tambahkan tumpukan StackSet dari menu Tindakan.
5. Pada halaman Setel opsi penerapan, pilih Impor tumpukan ke set tumpukan dan masukkan ID tumpukan tumpukan yang ingin Anda impor di bidang Tumpukan untuk diimpor. Misalnya, *arn:aws:cloudformation:us-east-1:123456789012:stack/StackToImport/f449b250-b969-11e0-a185-5081d0136786*.

(Opsional) Pilih Tambahkan ID tumpukan lain dan masukkan ID tumpukan tumpukan lain yang ingin Anda impor. Anda dapat menambahkan hingga 10 tumpukan per operasi impor tumpukan.

6. Pilih Berikutnya.
7. Pada halaman Tentukan penggantian, tinjau pilihan Anda dan pilih Berikutnya.
8. Pada halaman Ulasan, tinjau pilihan Anda StackSet dan properti Anda. Saat Anda siap untuk membuat StackSet, pilih Kirim.

Hasil: Tumpukan yang diimpor sekarang merupakan instance tumpukan yang ditentukan StackSet. Untuk mempelajari lebih lanjut tentang status impor tumpukan, lihat [StackSets kode status](#).

Impor tumpukan ke StackSet (AWS CLI)

Untuk mengimpor tumpukan yang ada ke yang baru StackSet

`create-stack-set` Perintah berikut membuat StackSet dan mengimpor tumpukan tertentu. Tumpukan yang akan diimpor diidentifikasi oleh ARN-nya. Ganti teks placeholder dengan informasi Anda sendiri.

```
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
```

Untuk mengimpor tumpukan yang ada ke dalam yang sudah ada StackSet

`import-stacks-to-stack-sets` Perintah berikut mengimpor tumpukan yang ditentukan ke dalam *MyStackSet* StackSet Tumpukan yang akan diimpor diidentifikasi oleh ARN-nya. Ganti teks placeholder dengan informasi Anda sendiri.

```
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
```

Untuk menentukan lebih dari satu tumpukan, gunakan format berikut untuk nilai `--stack-ids` opsi.

```
--stack-ids "arn_1" "arn_2"
```

Untuk mengkloning tumpukan yang diimpor ke Wilayah dan akun lain

`create-stack-instances` Perintah berikut menambahkan instance stack ke Anda StackSet. Ganti teks placeholder dengan informasi Anda sendiri.

```
aws cloudformation create-stack-instances \  
  --stack-set-name MyStackSet \  
  --accounts '["account_ID_1","account_ID_2"]' \  
  --stack-id arn:aws:cloudformation:us-east-1:123456789012:stack/StackToImport/  
f449b250-b969-11e0-a185-5081d0136786
```

```
--regions '["region_1","region_2"]'
```

Impor tumpukan yang dikelola layanan untuk AWS CloudFormation StackSets

Operasi impor CloudFormation tumpukan dapat mengimpor tumpukan yang ada ke yang baru atau yang sudah ada StackSets, sehingga Anda dapat memigrasikan tumpukan yang ada ke StackSet dalam satu operasi. StackSets memperluas fungsionalitas tumpukan, sehingga Anda dapat membuat, memperbarui, atau menghapus tumpukan di beberapa akun dan Wilayah dengan satu operasi.

Pertimbangan untuk impor tumpukan yang dikelola layanan

- Operasi impor tumpukan memerlukan akun manajemen atau akun admin yang didelegasikan tempat Anda dapat mengelola yang terkait AWS Organizations seperti mengaktifkan akses kepercayaan. StackSets
- Akun target harus menjadi anggota yang AWS Organizations dikelola oleh akun manajemen atau akun admin yang didelegasikan.
- Target stack ada di salah satu target OUs.
- Akun target harus menjadi anggota AWS Organizations.
- AWS Organizations akses harus di ACTIVATED negara bagian untuk Organizations.
- Tumpukan yang diimpor harus ada di salah satu akun anggota, dan bukan akun manajemen.

Topik

- [Impor tumpukan yang dikelola layanan ke StackSet \(konsol\) baru](#)
- [Membuat dan mengimpor tumpukan yang dikelola layanan ke StackSet \(konsol\) yang sudah ada](#)
- [Impor tumpukan yang dikelola layanan ke StackSet \(konsol\) yang sudah ada](#)
- [Mengimpor tumpukan yang dikelola layanan ke dalam \(\) StackSet AWS CLI](#)

Impor tumpukan yang dikelola layanan ke StackSet (konsol) baru

Impor tumpukan ke yang baru StackSet menggunakan AWS Management Console

Untuk mengimpor tumpukan baru ke dalam StackSet, identifikasi tumpukan yang berisi sumber daya yang ingin Anda impor.

1. Masuk ke AWS Management Console dan buka AWS CloudFormation konsol di <https://console.aws.amazon.com/cloudformation>.
2. Dari panel navigasi, pilih StackSets.
3. Di bagian atas StackSetshalaman, pilih Buat StackSet.
4. Pada halaman Choose a template, lakukan hal berikut:
 - a. Untuk model StackSet izin pilih Izin yang dikelola layanan.
 - b. Untuk Prasyarat - Siapkan template, pilih Template siap, dan pilih template Anda dengan menggunakan salah satu opsi berikut:
 - Untuk URL Amazon S3, masukkan URL Amazon S3 Anda di bidang URL Amazon S3.
 - Untuk Unggah file templat, pilih CloudFormation templat di komputer lokal Anda.

Terima pengaturan Anda dan pilih Berikutnya.

5. Pada halaman Tentukan StackSet detail, lakukan hal berikut:
 - a. Masukkan StackSet nama di kotak StackSet nama.
 - b. (Opsional) Masukkan deskripsi di bagian StackSet deskripsi.

Pada halaman Konfigurasi StackSet opsi, tinjau pilihan Anda dan pilih Berikutnya.

6. Pada halaman Set deployment options, lakukan hal berikut:
 - a. Untuk Tambahkan tumpukan ke set tumpukan, pilih Impor tumpukan ke set tumpukan.
 - b. Untuk Stacks untuk mengimpor, pilih metode impor tumpukan Anda.
 - i. Untuk Stack ID masukkan ID tumpukan Anda.
 - ii. Untuk URL Stack, masukkan URL Amazon S3.
7. Di bawah unit organisasi Associate, lakukan hal berikut:
 - a. Pilih Berasosiasi dengan organisasi untuk menggunakan root OU.
 - b. Pilih Kaitkan dengan unit organisasi (OUs) untuk memasukkan OU induk IDs agar tumpukan yang akan diimpor. Misalnya, jika Stack 1 dan Stack 2 berada di bawahOU1, dan Stack 3 berada di bawahOU2, masuk OU1 danOU2.

Terima pengaturan Anda dan pilih Berikutnya.

8. Tinjau pengaturan Anda di halaman Ulasan dan pilih Kirim.

Membuat dan mengimpor tumpukan yang dikelola layanan ke StackSet (konsol) yang sudah ada

Untuk mengimpor tumpukan yang ada ke yang baru StackSet, identifikasi tumpukan yang berisi sumber daya yang ingin Anda impor.

Untuk membuat StackSet dan mengimpor tumpukan

1. Masuk ke AWS Management Console dan buka AWS CloudFormation konsol di <https://console.aws.amazon.com/cloudformation>.
2. Dari panel navigasi, pilih StackSets.
3. Di bagian atas StackSetshalaman, pilih Buat StackSet.
4. Pada halaman Choose a template, lakukan hal berikut:
 - a. Untuk model StackSet izin pilih Izin yang dikelola layanan.
 - b. Untuk Prasyarat - Siapkan template, pilih Template siap, dan pilih template Anda dengan menggunakan salah satu opsi berikut:
 - Untuk URL Amazon S3, masukkan URL Amazon S3 Anda di bidang URL Amazon S3.
 - Untuk Unggah file templat, pilih CloudFormation templat di komputer lokal Anda.

Terima pengaturan Anda dan pilih Berikutnya.

5. Pada halaman Tentukan StackSet detail, lakukan hal berikut:
 - a. Masukkan StackSet nama di kotak StackSet nama.
 - b. (Opsional) Masukkan deskripsi di bagian StackSet deskripsi.

Pada halaman Konfigurasi StackSet opsi, tinjau pilihan Anda dan pilih Berikutnya.

6. Pada halaman Set deployment options, lakukan hal berikut:
 - Untuk Menambahkan stack ke set stack, pilih Terapkan stack baru.
7. Untuk bagian Unit organisasi asosiasi, lakukan hal berikut:
 - a. Pilih Berasosiasi dengan organisasi untuk menggunakan root OU.

- b. Pilih Kaitkan dengan unit organisasi (OUs) untuk memasukkan OU induk IDs agar tumpukan yang akan diimpor. Misalnya, jika Stack 1 dan Stack 2 berada di bawah OU1, dan Stack 3 berada di bawah OU2, masuk OU1 dan OU2.
8. Untuk opsi Tentukan wilayah dan Penerapan, tinjau pilihan Anda.

Terima pengaturan Anda dan pilih Berikutnya.

9. Tinjau pengaturan Anda di halaman Ulasan dan pilih Kirim.

Impor tumpukan yang dikelola layanan ke StackSet (konsol) yang sudah ada

Pilih StackSet dan identifikasi tumpukan yang ingin Anda impor.

Untuk mengimpor tumpukan ke yang sudah ada StackSet

1. Masuk ke AWS Management Console dan buka AWS CloudFormation konsol di <https://console.aws.amazon.com/cloudformation>.
2. Dari panel navigasi, pilih StackSets.
3. Pilih tumpukan yang ingin StackSet Anda impor, lalu pilih Tambahkan tumpukan StackSet dari menu tarik-turun Tindakan.
4. Pada halaman Set deployment options, lakukan hal berikut:
 - a. Untuk Tambahkan tumpukan ke set tumpukan, pilih Impor tumpukan ke set tumpukan.
 - b. Di bawah Tumpukan untuk mengimpor, lakukan hal berikut
 - i. Untuk Stack ID, masukkan ID tumpukan Anda.
 - ii. Untuk URL Stack, masukkan URL Amazon S3.
 - c. Di bawah unit organisasi Associate, lakukan hal berikut:
 - i. Pilih Berasosiasi dengan organisasi untuk menggunakan root OU.
 - ii. Pilih Kaitkan dengan unit organisasi (OUs) untuk memasukkan OU induk IDs agar tumpukan yang akan diimpor. Misalnya, jika Stack 1 dan Stack 2 berada di bawah OU1, dan Stack 3 berada di bawah OU2, masuk OU1 dan OU2.

Terima pengaturan Anda dan pilih Berikutnya.

5. Tinjau halaman Tentukan penggantian dan pilih Berikutnya.
6. Konfirmasikan dan tinjau halaman Ulasan dan pilih Kirim.

Mengimpor tumpukan yang dikelola layanan ke dalam () StackSet AWS CLI

Setelah StackSet dibuat, Anda dapat mengimpor tumpukan Anda dengan meneruskan ID tumpukan dari tumpukan yang diimpor. Anda juga dapat meneruskan daftar ID OU yang ingin Anda petakan.

CloudFormation akan mengimpor tumpukan yang disediakan pengguna di dalamnya OUs dan menggunakannya OUs sebagai target penerapan untuk. StackSet Tumpukan yang IDs disajikan dalam input akan memetakan ke OU terdekat dalam input daftar ID OU secara internal. Jika tumpukan bukan milik ID OU yang ada di daftar input, maka AWS CLI akan mengembalikan `StackNotFoundException` kesalahan.

`import-stacks-to-stack-set` Operasi membuat instance tumpukan untuk tumpukan di input ID OU. AWS CLI Contoh berikut menggunakan `import-stacks-to-stack-set` operasi untuk mengimpor tumpukan ke dalam StackSet.

- Untuk menggunakan `import-stacks-to-stack-sets` operasi, tentukan `stack-ids` atau `stack-ids-url` Anda ingin mengimpor ke set tumpukan Anda.

```
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

`import-stacks-to-stack-sets` Operasi, mengharuskan Anda untuk menentukan setidaknya satu ID unit organisasi (ID OU) sehingga dapat mengaitkan tumpukan yang diimpor ke OU tertentu. Operasi ini tidak membuat instance tumpukan untuk akun anggota lain di terkait OUs. Untuk memperbarui akun anggota untuk yang terkait OUs, menggunakan `create-stack-instances` atau `update-stack-instances`.

`create-stack-set` membuat instance tumpukan untuk semua akun di bawah template OUs yang disediakan pengguna, baik dari upload langsung atau Amazon S3. AWS CLI Contoh berikut menggunakan `create-stack-set` operasi untuk mengimpor tumpukan ke yang baru StackSet.

- Untuk menggunakan `create-stack-set` operasi, tentukan StackSet nama Anda dan impor tumpukan ke yang baru dibuat 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
```

Kembalikan impor tumpukan ke AWS CloudFormation StackSets

Jika Anda memiliki perubahan yang tidak diinginkan dalam instance stack Anda, Anda dapat mengembalikan impor instance stack.

Saat Anda mengembalikan impor instance tumpukan, CloudFormation menghapus instance tumpukan dari StackSet tetapi mempertahankan sumber daya tumpukan.

Untuk mengembalikan operasi impor tumpukan, selesaikan prosedur berikut.

1. Tentukan `DeletionPolicy` atribut `Retain` untuk setiap sumber daya yang ingin Anda simpan setelah instance tumpukan dihapus. Untuk informasi selengkapnya, lihat [Pengembalian operasi impor](#).
2. Hapus instance tumpukan dari Anda StackSet. Untuk informasi selengkapnya, lihat [Hapus tumpukan dari AWS CloudFormation StackSets](#).
3. Hapus Anda StackSet. Lihat informasi yang lebih lengkap di [Hapus AWS CloudFormation StackSets](#).

Praktik terbaik untuk menggunakan AWS CloudFormation StackSets

Bagian ini menjelaskan praktik terbaik untuk mendefinisikan StackSet templat, membuat atau menambahkan tumpukan ke StackSet, atau memperbarui templat. StackSet

Jika Anda baru mengenal CloudFormation, tinjau [CloudFormation praktik terbaik](#) topik untuk rekomendasi lebih lanjut yang dapat membantu Anda menggunakan dengan CloudFormation lebih efektif dan aman.

Topik

- [Penentuan templat](#)
- [Membuat atau menambahkan tumpukan ke StackSet](#)
- [Memperbarui tumpukan di StackSet](#)

Penentuan templat

- Tentukan templat yang ingin Anda standarisasi di beberapa akun, dalam beberapa wilayah.
- Saat Anda membuat template, pastikan bahwa sumber daya global (seperti peran IAM dan bucket Amazon S3) tidak memiliki konflik penamaan saat dibuat di lebih dari satu wilayah di akun yang sama.
- A StackSet memiliki satu template dan set parameter. Tumpukan yang sama dibuat di semua akun yang terkait dengan file StackSet. Saat Anda menulis templat Anda, buatlah templat tersebut cukup terperinci untuk memungkinkan Anda keseimbangan kontrol dan standardisasi yang baik.
- Kami menyarankan Anda menyimpan template Anda di ember Amazon S3.

Membuat atau menambahkan tumpukan ke StackSet

- Verifikasi bahwa menambahkan instance tumpukan ke StackSet pekerjaan awal Anda sebelum Anda menambahkan jumlah instance tumpukan yang lebih besar ke Anda. StackSet
- Pilih pilihan deployment (peluncuran) yang berfungsi untuk kasus penggunaan Anda.
 - Untuk deployment yang lebih konservatif, atur Akun Konkuren Maksimum ke 1, dan Toleransi Kegagalan ke 0. Atur wilayah dampak terendah Anda menjadi yang pertama di daftar Urutan Wilayah. Mulailah dengan satu wilayah.
 - Untuk deployment yang lebih cepat, tingkatkan nilai AkunKonkuren Maksimum dan Toleransi Kegagalan sesuai kebutuhan.
- Operasi StackSets tergantung pada berapa banyak instance tumpukan yang terlibat, dan dapat memakan waktu yang signifikan.

Memperbarui tumpukan di StackSet

- Secara default, memperbarui StackSet pembaruan semua instance tumpukan. Jika Anda memiliki 20 akun masing-masing di dua wilayah, Anda akan memiliki 40 instance tumpukan, dan semuanya akan diperbarui saat Anda memperbarui StackSet

Karena StackSets dengan sejumlah besar instance tumpukan, kami menyarankan agar untuk menguji versi template yang diperbarui, Anda secara selektif memperbarui instance tumpukan di beberapa akun pengujian sebelum memperbarui semua instance tumpukan.

- Untuk mendapatkan kontrol yang lebih terperinci atas memperbarui tumpukan individual di dalam Anda StackSet, rencanakan untuk membuat beberapa StackSets
- Memperbarui StackSet yang berisi sejumlah besar tumpukan dapat memakan waktu yang signifikan. Dalam rilis ini, hanya satu operasi yang diizinkan pada satu waktu pada a StackSet. Rencanakan pembaruan Anda sehingga Anda tidak diblokir untuk melakukan operasi lain di StackSet.

AWS CloudFormation StackSets contoh template

Bagian ini mencakup tautan ke beberapa contoh CloudFormation templat yang dapat membantu Anda gunakan AWS CloudFormation StackSets di perusahaan Anda. Templat yang tercantum di bagian ini mengaktifkan [AWS CloudTrail](#) atau [AWS Config](#) dan aturan di dalamnya.

Important

Sebagai praktik keamanan terbaik saat mengizinkan AWS Config akses ke bucket Amazon S3, kami sangat menyarankan Anda membatasi akses dalam kebijakan bucket dengan kondisi tersebut. `AWS:SourceAccount` Template baru diperbarui untuk memiliki `AWS:SourceAccount`. Jika kebijakan bucket Anda yang ada tidak mengikuti praktik terbaik keamanan ini, kami sangat menyarankan Anda mengedit kebijakan bucket tersebut untuk menyertakan perlindungan ini. Hal ini memastikan bahwa AWS Config diberikan akses atas nama pengguna yang diharapkan saja.

Deskripsi	Tautan S3
Aktifkan AWS CloudTrail	https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/AktifkanAWSCloudtrail .yml.tl
Aktifkan AWS Config	https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/AktifkanAWSConfig .yml.tl
Aktifkan AWS Config dengan logging pusat	https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/AktifkanAWSConfig ForOrganizations .yml.tl
Mengaktifkan kebijakan default Amazon Data Lifecycle Manager di seluruh AWS organisasi atau di seluruh akun tertentu AWS	• Kebijakan default untuk snapshot EBS — .yaml https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/DataLifecycleManagerEBSSnapshotDefaultPolicy • Kebijakan default untuk EBS-Backed — AMIs Policy.yaml 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
Konfigurasi AWS Config aturan untuk menentukan CloudTrail apakah diaktifkan	https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/ConfigRuleCloudtrailEnabled.yml
Konfigurasi AWS Config aturan untuk menentukan apakah MFA root diaktifkan	https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/ConfigRuleRootAccountMFAEnabled.yml

Deskripsi	Tautan S3
Konfigurasi AWS Config aturan untuk menentukan apakah EIPs terlampir	https://s3.amazonaws.com/cloudformation-stacks/sample-templates-us-east-1/ConfigRuleEipAttached.yml
Konfigurasi AWS Config aturan untuk menentukan apakah volume EBS dienkripsi	https://s3.amazonaws.com/cloudformation-stacks/sample-templates-us-east-1/ConfigRuleEncryptedVolumes.yml

Pemecahan masalah AWS CloudFormation StackSets

Topik ini berisi beberapa StackSets masalah umum, dan solusi yang disarankan untuk masalah tersebut.

Topik

- [Alasan umum untuk kegagalan operasi tumpukan](#)
- [Mencoba lagi pembuatan tumpukan atau operasi pembaruan yang gagal](#)
- [Penghapusan instans tumpukan gagal](#)
- [Operasi impor tumpukan gagal](#)
- [Jumlah kegagalan instance stack untuk StackSets operasi](#)

Alasan umum untuk kegagalan operasi tumpukan

Masalah: Operasi tumpukan gagal, dan status instans tumpukan adalah OUTDATED.

Penyebab: Ada beberapa penyebab umum kegagalan operasi tumpukan.

- Izin tidak memadai di akun target untuk membuat sumber daya yang ditentukan di templat Anda.
- CloudFormation Template mungkin memiliki kesalahan. Validasi template CloudFormation dan perbaiki kesalahan sebelum mencoba membuat template Anda StackSet.
- Templat dapat mencoba untuk membuat sumber daya global yang harus unik tetapi tidak, seperti bucket S3.
- Nomor akun target yang ditentukan tidak ada. Periksa nomor akun target yang Anda tentukan di halaman Mengatur opsi deployment dari panduan.

- Akun administrator tidak memiliki hubungan kepercayaan dengan akun target.
- Jumlah maksimum sumber daya yang ditentukan dalam templat Anda sudah ada di akun target. Misalnya, Anda mungkin telah mencapai batas peran IAM yang diizinkan di akun target, tetapi template membuat lebih banyak peran IAM.
- Anda telah mencapai jumlah maksimum tumpukan yang diizinkan dalam a StackSet. Untuk jumlah maksimum tumpukan per StackSet, lihat[Memahami CloudFormation kuota](#).

Solusi: Untuk informasi selengkapnya tentang izin yang diperlukan dari akun target dan administrator sebelum Anda dapat membuat StackSets, lihat[Berikan semua pengguna izin akun administrator untuk mengelola tumpukan di semua akun target](#).

Mencoba lagi pembuatan tumpukan atau operasi pembaruan yang gagal

Masalah: Pembuatan atau pembaruan tumpukan gagal, dan status instans tumpukan adalah OUTDATED. Untuk memecahkan masalah mengapa pembuatan atau pembaruan tumpukan gagal, buka CloudFormation konsol, dan lihat peristiwa untuk tumpukan, yang akan memiliki status DELETED (untuk operasi pembuatan yang gagal) atau FAILED (untuk operasi pembaruan yang gagal). Telusuri peristiwa tumpukan, dan cari kolom Alasan status. Nilai dari Alasan status menjelaskan alasan operasi tumpukan mengalami kegagalan.

Setelah Anda memperbaiki penyebab utama kegagalan pembuatan tumpukan, dan siap untuk mencoba lagi pembuatan tumpukan, lakukan langkah-langkah berikut.

Solusi: Lakukan langkah-langkah berikut untuk mencoba lagi operasi tumpukan.

1. Di konsol, pilih StackSet yang berisi tumpukan tempat operasi gagal.
2. Di menu Tindakan, pilih Edit StackSet detail untuk mencoba lagi membuat atau memperbarui tumpukan.
3. Di halaman Tentukan templat, untuk menggunakan templat CloudFormation yang sama, simpan opsi default, Gunakan templat saat ini. Jika operasi tumpukan Anda gagal karena templat memerlukan perubahan, dan Anda ingin mengunggah templat yang direvisi, pilih Unggah templat ke Amazon S3, lalu pilih Telusuri untuk memilih templat yang diperbarui. Setelah selesai mengunggah templat yang direvisi, pilih Selanjutnya.
4. Pada halaman Tentukan detail tumpukan, jika Anda tidak mengubah parameter apa pun yang spesifik untuk templat Anda, pilih Berikutnya.

5. Di halaman Atur opsi deployment, ubah default untuk Akun serentak maksimum dan Toleransi kegagalan, jika diinginkan. Untuk informasi selengkapnya tentang pengaturan ini, lihat [StackSet opsi operasi](#).
6. Pada halaman Tinjauan, tinjau pilihan Anda, dan isi kotak centang untuk mengetahui kemampuan IAM yang diperlukan. Pilih Kirim.
7. Jika tumpukan Anda tidak berhasil diperbarui, ulangi prosedur ini, setelah Anda menyelesaikan masalah mendasar yang mencegah pembuatan tumpukan.

Penghapusan instans tumpukan gagal

Masalah: Penghapusan tumpukan telah mengalami kegagalan.

Penyebab: Penghapusan tumpukan akan gagal untuk setiap tumpukan di mana perlindungan pengakhiran telah diaktifkan.

Solusi: Tentukan apakah perlindungan pengakhiran telah diaktifkan untuk tumpukan tersebut. Jika sudah, nonaktifkan perlindungan pengakhiran, kemudian lakukan penghapusan instans tumpukan lagi.

Operasi impor tumpukan gagal

Masalah: Operasi impor tumpukan gagal mengimpor tumpukan yang ada ke yang baru atau yang sudah ada StackSets. Instans tumpukan berada dalam INOPERABLE status.

Solusi: Kembalikan operasi impor tumpukan, dengan menyelesaikan tugas-tugas berikut.

1. Gunakan Hapus Tumpukan dari StackSets opsi dan aktifkan RetainStacksselama konfigurasi, lalu lanjutkan untuk menghapus instance tumpukan dari Anda. StackSet Untuk informasi lebih lanjut, Untuk informasi lebih lanjut, lihat [Hapus tumpukan dari AWS CloudFormation StackSets](#) .
2. Anda akan melihat instance tumpukan yang StackSet diperbarui untuk menghapus instance INOPERABLE tumpukan.
3. Perbaiki instance tumpukan sesuai dengan kesalahan kegagalan impor dan coba lagi operasi impor tumpukan.

Jumlah kegagalan instance stack untuk StackSets operasi

Jumlah kegagalan instans tumpukan memberi tahu Anda jika instance tumpukan gagal disediakan atau diperbarui. Instans tumpukan ini tidak diterapkan karena satu atau lebih alasan berikut:

- Sumber daya yang ada dengan konfigurasi serupa
- Dependensi yang hilang, seperti peran AWS Identity and Access Management (IAM)
- Faktor lain yang saling bertentangan

Jika Anda ingin menerapkan dengan konkurensi maksimum, jumlah konkurensi maks adalah satu lebih banyak dari jumlah toleransi kegagalan, paling banyak. Misalnya, jika jumlah toleransi kegagalan adalah 9, maka jumlah konkurensi maks tidak boleh lebih dari 10. Ini akan menyebabkan operasi kembali SUCCEEDED bahkan jika beberapa instance tumpukan gagal diperbarui. Jumlah kegagalan instans tumpukan baru memungkinkan Anda menentukan apakah operasi hanya berhasil secara kondisional karena jumlah toleransi kegagalan diatur untuk memungkinkan semua kegagalan.

Anda dapat menggunakan AWS Management Console, AWS SDK, atau AWS CLI untuk mendapatkan hitungan kegagalan dan memfilter instance tumpukan untuk menentukan instance mana yang perlu di-deploy ulang.

Menggunakan konsol

Untuk melihat jumlah instance tumpukan yang gagal:

1. Buka [CloudFormationkonsol](#) dan pilih StackSets.
2. Pilih tab Anda StackSet, lalu pilih tab Operasi.
3. Pilih status di kolom Status untuk melihat detail status. Anda akan menemukan jumlah instance tumpukan gagal untuk operasi tertentu dalam detail status.

Untuk melihat akun, wilayah, dan status instance tumpukan untuk operasi:

1. Dalam detail status, pilih jumlah instans tumpukan yang gagal. Contoh: Stack instance: **<number of failed stack instances>**.
2. Perluas panel samping dengan memilih header panel. Hasil di panel samping adalah status instance tumpukan setelah selesainya operasi yang dipilih.

Untuk melihat detail instance tumpukan saat ini untuk operasi:

1. Pilih tab Stack Instances.
2. Filter berdasarkan ID operasi terakhir. Hasilnya adalah status dan alasan status saat ini dari operasi terakhir untuk memodifikasi instance. Anda dapat menggunakan filter ini dalam

kombinasi dengan AWS akun, AWS wilayah, Status Terperinci, dan status Drift untuk lebih menyempurnakan hasil pencarian Anda.

Menggunakan AWS CLI

Untuk mendapatkan jumlah instance tumpukan yang gagal, panggil `describe-stack-set-operation` atau `list-stack-set-operations` dan lihat `StatusDetails`.

```
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
        }
    }
]
}

```

Untuk mendapatkan gambaran historis untuk operasi tertentu, gunakan `list-stack-set-operation-results` untuk melihat status dan alasan status untuk setiap instance tumpukan setelah operasi selesai. Lihat contoh berikut untuk menemukan `Status` dan `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": ""
    }
]
}

```

Gunakan `list-stack-instances` dengan `LAST_OPERATION_ID` filter `DETAILED_STATUS` dan untuk mendapatkan daftar instance tumpukan yang gagal dalam operasi terakhir yang mencoba menerapkan instance tumpukan. Lihat `--filters` bendera dalam contoh dengan `DETAILED_STATUS` dan `LAST_OPERATION_ID`:

```

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"
}

```

```

    }
  ]
}

```

Untuk menemukan ID operasi terakhir untuk memodifikasi instance tumpukan, gunakan `list-stack-instances` atau `describe-stack-instance` untuk mendapatkan `LastOperationId`:

```
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"
  }
}
```

Menyinkronkan tumpukan dengan kode sumber yang disimpan dalam repositori Git dengan sinkronisasi Git

Dengan sinkronisasi Git, Anda dapat mengelola CloudFormation tumpukan Anda dengan kontrol sumber. Anda melakukan ini dengan mengkonfigurasi CloudFormation untuk memantau repositori Git. Repositori dipantau untuk perubahan pada dua file:

- File CloudFormation template yang mendefinisikan tumpukan
- File penyebaran tumpukan yang berisi parameter yang mengonfigurasi tumpukan

Dengan sinkronisasi Git, Anda dapat menggunakan permintaan tarik dan pelacakan versi untuk mengonfigurasi, menyebarkan, dan memperbarui CloudFormation tumpukan Anda dari lokasi terpusat. Saat Anda melakukan perubahan pada template atau file penerapan, CloudFormation secara otomatis memperbarui tumpukan. Jika Anda menggunakan permintaan tarik, CloudFormation dapat meninggalkan komentar pada permintaan tarik yang menjelaskan perubahan apa yang akan dilakukan pada tumpukan Anda sebelum benar-benar memperbaruinya. Namun, Anda harus mengaktifkan fitur ini terlebih dahulu.

Git sync menyediakan antarmuka konsol yang dapat Anda gunakan untuk menautkan ke repositori, menghasilkan file penyebaran tumpukan, memperbarui CloudFormation template, dan mengirimkan permintaan tarik ke repositori Anda. Sinkronisasi Git juga menyediakan dasbor status yang dapat Anda gunakan untuk memantau, mengedit, dan memecahkan masalah penerapan tumpukan sinkronisasi Git yang aktif. Sinkronisasi Git diakses melalui [CloudFormation konsol](#) saat Anda [membuat tumpukan](#). Anda juga dapat mengakses sinkronisasi Git menggunakan CodeConnections. Untuk informasi selengkapnya, lihat [Bekerja dengan konfigurasi sinkronisasi untuk repositori tertaut di Panduan Pengguna Konsol Alat Pengembang](#).

Git sync mendukung [GitHub](#), [GitHub Enterprise](#), [Bitbucket](#) [GitLab](#), dan repositori yang [GitLab dikelola sendiri](#).

Note

Git sync tersedia di wilayah berikut: AS Timur (Virginia N.), AS Timur (Ohio), AS Barat (California N.), AS Barat (Oregon), Kanada (Tengah), Asia Pasifik (Mumbai), Asia Pasifik (Tokyo), Asia Pasifik (Seoul), Asia Pasifik (Singapura), Asia Pasifik (Sydney), Eropa

(Irlandia), Eropa (London), Eropa (London), Eropa (Paris), Eropa (Stockholm), Eropa (Frankfurt), Eropa (Milan), dan Amerika Selatan (São Paulo).

Untuk informasi tentang penggunaan sinkronisasi Git dengan strategi multi-akun, lihat posting blog berikut [Gunakan sinkronisasi AWS CloudFormation Git untuk mengonfigurasi sumber daya di akun pelanggan](#).

Topik

- [Bagaimana sinkronisasi Git bekerja dengan CloudFormation](#)
- [Prasyarat untuk menyinkronkan tumpukan ke repositori Git menggunakan sinkronisasi Git](#)
- [Buat tumpukan dari kode sumber repositori dengan sinkronisasi Git](#)
- [Aktifkan CloudFormation untuk memposting ringkasan perubahan tumpukan dalam permintaan tarik](#)
- [Dasbor status sinkronisasi Git](#)

Bagaimana sinkronisasi Git bekerja dengan CloudFormation

Topik ini menjelaskan cara kerja sinkronisasi Git dan memperkenalkan konsep-konsep kunci yang diperlukan untuk bekerja dengannya.

Topik

- [Cara kerja sinkronisasi Git](#)
- [Komentar pada permintaan tarik](#)
- [File penyebaran tumpukan](#)
- [CloudFormation berkas templat](#)
- [Repositori definisi templat](#)

Cara kerja sinkronisasi Git

Untuk menggunakan sinkronisasi Git, pertama-tama Anda harus menghubungkan penyedia Git untuk CloudFormation menggunakan [CodeConnections](#) layanan. Dalam prosedur dalam panduan ini, koneksi dibuat melalui CodeConnections konsol. Atau, Anda dapat membuat koneksi dengan AWS CLI. Anda dapat menggunakan salah satu penyedia Git berikut:

- [GitHub](#)
- [GitHub Perusahaan](#)
- [GitLab](#)
- [Bitbucket](#)
- [GitLabdikelola sendiri](#)

Selanjutnya, Anda membuat CloudFormation template yang mendefinisikan tumpukan Anda dan menambahkannya ke repositori Anda. File template ini dipantau. CloudFormation memperbarui tumpukan secara otomatis saat perubahan dilakukan padanya.

Di CloudFormation konsol, Anda membuat tumpukan baru dan memilih Sinkronkan dari Git untuk memberitahu CloudFormation agar menggunakan sinkronisasi Git. Anda akan menentukan repositori dan cabang yang CloudFormation ingin Anda pantau, dan menentukan CloudFormation template di repositori Anda yang mendefinisikan tumpukan.

Selama konfigurasi, Anda dapat menyediakan file penyebaran tumpukan Anda sendiri dari repositori Anda, atau membuat sinkronisasi Git untuk Anda. File penyebaran tumpukan berisi parameter dan nilai yang mengonfigurasi sumber daya di tumpukan Anda. File penyebaran tumpukan ini dipantau. CloudFormation memperbarui tumpukan secara otomatis saat perubahan dilakukan padanya.

Sinkronisasi Git membuat permintaan tarik di repositori Anda untuk menyinkronkan tumpukan Anda dengan file CloudFormation template dan file penyebaran tumpukan. Jika Git sync menghasilkan file penyebaran tumpukan untuk Anda, itu dikirimkan ke repositori Anda oleh sinkronisasi Git.

Anda kemudian menggabungkan permintaan tarik ke repositori Anda sehingga CloudFormation menyediakan tumpukan, mengonfigurasinya dengan parameter penerapan Anda, dan mulai memantau repositori Anda untuk perubahan.

Sejak saat itu, setiap kali Anda membuat perubahan pada file template atau file penyebaran tumpukan dan memasukkannya ke repositori Anda, CloudFormation akan secara otomatis mendeteksi perubahan. Jika tim Anda menggunakan permintaan tarik, anggota tim Anda kemudian dapat meninjau dan menyetujui perubahan sebelum diterapkan. Setelah permintaan tarik diterima, CloudFormation gunakan perubahan Anda.

Anda dapat memantau status konfigurasi sinkronisasi Git Anda untuk tumpukan dan melihat riwayat komit yang diterapkan ke tumpukan di CloudFormation konsol. Konsol ini juga menyediakan alat untuk mengonfigurasi ulang masalah sinkronisasi Git dan pemecahan masalah.

Komentar pada permintaan tarik

Anda dapat memilih untuk CloudFormation membuat ringkasan perubahan kode dalam permintaan tarik melalui CodeConnections layanan dengan mengaktifkan opsi Aktifkan komentar pada permintaan tarik di konsol. Memberikan ringkasan perubahan permintaan tarik berarti anggota tim dapat dengan mudah meninjau dan memahami dampak modifikasi yang diusulkan sebelum menggabungkan permintaan tarik. Untuk informasi selengkapnya, lihat [Aktifkan CloudFormation untuk memposting ringkasan perubahan tumpukan dalam permintaan tarik](#).

File penyebaran tumpukan

File penyebaran tumpukan adalah file berformat standar JavaScript Object Notation (JSON) atau YAMAL yang berisi parameter dan nilai yang mengelola tumpukan Anda. CloudFormation itu dipantau untuk perubahan. Ketika perubahan pada file berkomitmen ke repositori, tumpukan terkait diperbarui secara otomatis.

File penyebaran tumpukan berisi pasangan kunci-nilai dan dua kamus:

- `template-file-path`

Ini adalah jalur repositori lengkap untuk file CloudFormation template. File template mendeklarasikan sumber daya untuk CloudFormation tumpukan yang terkait dengan file penerapan ini.

- `parameters`

Kamus parameter berisi pasangan kunci-nilai yang mengonfigurasi sumber daya dalam tumpukan. File penyebaran tumpukan dapat memiliki hingga 50 parameter.

- `tags`

Kamus tag berisi pasangan nilai kunci opsional yang dapat Anda gunakan untuk mengidentifikasi dan mengkategorikan sumber daya dalam tumpukan. File penyebaran tumpukan dapat memiliki hingga 50 tag.

Anda dapat menyediakan file penyebaran tumpukan Anda sendiri, atau membuat sinkronisasi Git untuk Anda dan secara otomatis mengirimkan permintaan tarik ke repositori Anda. Anda dapat mengelola parameter dan tag dengan mengedit file penyebaran tumpukan dan melakukan perubahan pada repositori.

Berikut ini adalah contoh file penyebaran tumpukan sinkronisasi Git:

```
template-file-path: fargate-srvs/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 berkas templat

File template berisi deklarasi AWS sumber daya yang membentuk CloudFormation tumpukan. Dengan sinkronisasi Git, file template disimpan di repositori Git Anda dan direferensikan oleh file penyebaran tumpukan. Anda dapat mengelola tumpukan dengan mengedit file template dan melakukan perubahan pada repositori.

Untuk informasi selengkapnya, lihat [Bekerja dengan CloudFormation template](#).

Repositori definisi templat

Repositori definisi template adalah repositori Git yang ditautkan melalui CloudFormation sinkronisasi Git. Repositori dipantau untuk perubahan pada CloudFormation template dan file penyebaran tumpukan. Saat Anda melakukan perubahan pada file, tumpukan terkait diperbarui secara otomatis.

Important

Saat Anda mengonfigurasi repositori definisi template di konsol sinkronisasi Git, pilih repositori dan cabang yang benar dari koneksi Git. Git sync hanya memantau repositori dan cabang yang dikonfigurasi untuk perubahan pada CloudFormation template dan file penyebaran tumpukan.

Prasyarat untuk menyinkronkan tumpukan ke repositori Git menggunakan sinkronisasi Git

Sebelum Anda menyinkronkan CloudFormation tumpukan ke repositori Git Anda, verifikasi bahwa persyaratan berikut terpenuhi.

Topik

- [Repositori Git](#)
- [CloudFormation Template](#)
- [Peran IAM](#)
- [Izin yang diperlukan](#)
- [Kebijakan kepercayaan](#)

Repositori Git

Anda harus memiliki repositori Git yang dihosting di salah satu platform berikut:

- [GitHub](#)
- [GitHub Perusahaan](#)
- [GitLab](#)
- [Bitbucket](#)
- [GitLabdikelola sendiri](#)

Repositori dapat bersifat publik atau pribadi. Anda harus menghubungkan repositori Git ini ke CloudFormation konsol [Connections](#).

CloudFormation Template

Repositori Git Anda harus berisi [file CloudFormation template](#) yang diperiksa ke cabang yang ingin Anda hubungkan dengan sinkronisasi Git. Template ini akan direferensikan oleh file [stack deployment](#).

Peran IAM

Sinkronisasi Git membutuhkan peran IAM. Anda dapat memilih untuk memiliki peran IAM yang dibuat untuk tumpukan Anda ketika Anda mengkonfigurasi sinkronisasi Git, atau Anda dapat menggunakan peran yang ada.

Note

Peran IAM yang dihasilkan secara otomatis hanya menerapkan izin ke tumpukan tempat peran tersebut dihasilkan. Untuk menggunakan kembali peran IAM yang dibuat secara otomatis, Anda harus mengedit peran untuk tumpukan baru.

Izin yang diperlukan

Peran IAM yang Anda berikan untuk sinkronisasi Git memerlukan izin berikut:

- [cloudformation:CreateChangeSet](#)
- [cloudformation>DeleteChangeSet](#)
- [cloudformation:DescribeChangeSet](#)
- [cloudformation:DescribeStackEvents](#)
- [cloudformation:DescribeStacks](#)
- [cloudformation:ExecuteChangeSet](#)
- [cloudformation:ListChangeSets](#)
- [cloudformation:ValidateTemplate](#)
- [events:PutRule](#)
- [events:PutTargets](#)

Note

Izin yang diperlukan sebelumnya secara otomatis ditambahkan ke peran IAM yang dihasilkan oleh sinkronisasi Git.

Contoh peran IAM berikut mencakup izin prasyarat untuk sinkronisasi Git.

```

{
  "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": "*"
    }
  ]
}

```

Kebijakan kepercayaan

Kebijakan kepercayaan berikut diperlukan untuk sinkronisasi Git untuk terhubung ke repositori Git Anda:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CfnGitSyncTrustPolicy",
      "Effect": "Allow",
      "Principal": {
        "Service": "cloudformation.sync.codeconnections.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Buat tumpukan dari kode sumber repositori dengan sinkronisasi Git

Topik ini menjelaskan cara membuat CloudFormation tumpukan yang disinkronkan ke repositori Git dengan sinkronisasi Git.

Important

Sebelum Anda melanjutkan, lengkapi semua [prasyarat di bagian](#) sebelumnya.

Buat tumpukan dari kode sumber repositori

1. Masuk ke AWS Management Console dan buka AWS CloudFormation konsol di <https://console.aws.amazon.com/cloudformation>.
2. Pada bilah navigasi di bagian atas layar, pilih Wilayah AWS untuk membuat tumpukan.
3. Pada halaman Stacks, pilih Buat tumpukan, lalu pilih Dengan sumber daya baru (standar).
4. Pada halaman Buat tumpukan, lakukan hal berikut:
 - a. Untuk Prasyarat - Siapkan templat, simpan Pilih templat yang ada yang dipilih.
 - b. Untuk template Tentukan, pilih Sinkronkan dari Git, lalu pilih Berikutnya.

5. Pada halaman Tentukan detail tumpukan, untuk nama Stack, ketikkan nama untuk tumpukan Anda. Nama tumpukan dapat mencakup huruf (A-Z dan a-z), angka (0-9), dan tanda hubung (-).
6. Untuk file penyebaran Stack, pembuatan file Deployment:
 - Jika Anda belum membuat file penyebaran tumpukan dan menambahkannya ke repositori Anda, pilih Buat file menggunakan parameter berikut dan letakkan di repositori saya.
 - Jika Anda memiliki file penyebaran tumpukan di repositori Anda, pilih Saya menyediakan file saya sendiri di repositori saya.
7. Untuk repositori definisi Template, pilih Pilih repositori Git tertaut untuk memilih repositori Git yang sudah ditautkan ke, CloudFormation atau Tautkan repositori Git untuk menautkan yang baru. Jika Anda memilih Tautkan repositori Git, lakukan hal berikut:
 - a. Untuk Pilih penyedia repositori, pilih salah satu dari berikut ini:
 - GitHub
 - GitHub Server Perusahaan
 - GitLab
 - Bitbucket
 - GitLab dikelola sendiri
 - b. Untuk Koneksi, pilih koneksi dari daftar. Jika tidak ada opsi yang muncul di daftar Koneksi, pilih tambahkan koneksi baru untuk pergi ke [konsol Koneksi](#) dan buat koneksi ke repositori Anda.
8. Dalam daftar Repositori, pilih repositori Git yang berisi file template stack Anda.
9. Dalam daftar Branch, pilih cabang yang Anda ingin sinkronisasi Git untuk dipantau.

 Note

Sinkronisasi Git hanya memantau cabang yang dipilih untuk perubahan pada CloudFormation template dan file penyebaran tumpukan. Setiap perubahan yang ingin Anda terapkan ke tumpukan Anda harus berkomitmen ke cabang ini.

10. Untuk jalur file Deployment, tentukan jalur lengkap termasuk nama file penyebaran tumpukan dari root cabang repositori Anda.

Jika CloudFormation menghasilkan file untuk Anda, di sinilah file akan dikomit di repositori Anda. Jika Anda menyediakan file, ini adalah lokasi file di repositori Anda.

11. Tambahkan peran IAM. Peran IAM mencakup izin yang diperlukan untuk CloudFormation menyinkronkan tumpukan dari repositori Git Anda. Anda dapat memilih peran IAM baru untuk menghasilkan peran baru, atau memilih peran IAM yang ada untuk memilih peran yang ada dari peran Anda. Akun AWS Jika Anda memilih untuk membuat peran baru, izin yang diperlukan akan disertakan dalam peran.
12. Aktifkan atau matikan komentar pada permintaan tarik:
 - Agar informasi set perubahan CloudFormation posting dalam permintaan tarik untuk pembaruan tumpukan, aktifkan sakelar Aktifkan komentar pada permintaan tarik.
 - Jika Anda menonaktifkan sakelar ini, CloudFormation tidak akan menjelaskan perbedaan antara konfigurasi tumpukan saat ini dan perubahan yang diusulkan dalam permintaan tarik saat file repo diperbarui.
13. Untuk path file Template, tentukan path lengkap dari root repositori Anda untuk file template stack.
14. (Opsional) Untuk menentukan parameter tumpukan, pilih Tambah parameter, berikan kunci dan nilai untuk setiap parameter, lalu pilih Berikutnya. Untuk informasi selengkapnya, lihat [File penyebaran tumpukan](#).

Misalnya, untuk menentukan **port=8080** parameter dalam file penyebaran tumpukan Anda, lakukan hal berikut:

- a. Pilih Add parameter (Tambahkan parameter).
 - b. Untuk Kunci, masukkan **port**.
 - c. Untuk Nilai, masukkan **8080**.
15. (Opsional) Untuk menentukan tag tumpukan, pilih Tambahkan tag baru, berikan kunci tag dan nilai untuk setiap tag, lalu pilih Berikutnya. Untuk informasi selengkapnya, lihat [File penyebaran tumpukan](#).
 16. Pilih Berikutnya untuk melanjutkan ke Configure stack options. Untuk informasi tentang mengonfigurasi opsi tumpukan, lihat [Konfigurasi opsi tumpukan](#).

Ketika Anda telah menyelesaikan konfigurasi tumpukan Anda, pilih Berikutnya untuk melanjutkan.

17. Tinjau pengaturan tumpukan Anda dan konfirmasi hal berikut:
 - Template tumpukan dikonfigurasi dengan benar dan disetel ke Sync dari Git.
 - File penyebaran dikonfigurasi dengan benar.

- Repositori definisi template dikonfigurasi dengan benar, khususnya, bahwa nama Repositori dan Cabang yang benar dipilih.
- Pratinjau file penerapan sudah benar dan berisi parameter dan nilai yang diharapkan.

18. Pilih Kirim untuk membuat tumpukan.

Setelah Anda memilih Kirim, permintaan tarik secara otomatis dibuat di repositori Git Anda. Anda harus menggabungkan permintaan tarik ini ke dalam repositori Git Anda untuk membuat tumpukan Anda. Setelah tumpukan dibuat, CloudFormation monitor repositori Git Anda untuk perubahan.

Perbarui tumpukan Anda dari repositori Git Anda

Untuk memperbarui tumpukan, buat perubahan langsung ke file template Anda atau file penyebaran tumpukan di repositori Git Anda. Setelah Anda melakukan perubahan ke cabang yang dipantau, CloudFormation secara otomatis memperbarui tumpukan. Jika Anda menggunakan permintaan tarik, permintaan tarik secara otomatis dibuat di repositori Git Anda sebelum tumpukan diperbarui. Anda harus menggabungkan permintaan tarik ini ke dalam repositori Git Anda untuk memperbarui tumpukan Anda.

Di CloudFormation konsol, Anda dapat memilih tumpukan dan memilih tab sinkronisasi Git untuk melihat informasi tentang status tumpukan dan peristiwa sinkronisasi. Untuk informasi selengkapnya, lihat [Dasbor status sinkronisasi Git](#).

Aktifkan CloudFormation untuk memposting ringkasan perubahan tumpukan dalam permintaan tarik

Topik ini menunjukkan kepada Anda cara mengaktifkan CloudFormation untuk memposting ringkasan perubahan tumpukan dalam permintaan tarik di repositori Git Anda.

Dengan mengaktifkan fitur komentar pada permintaan tarik, Anda mengizinkan CloudFormation untuk memposting komentar yang menjelaskan perbedaan antara konfigurasi tumpukan saat ini dan perubahan yang diusulkan saat file repo diperbarui. Komentar ini memberikan ringkasan sumber daya yang akan ditambahkan, dimodifikasi, atau dihapus, memungkinkan Anda untuk melakukan tinjauan kode menyeluruh sebelum menggabungkan permintaan tarik.

Untuk mengaktifkan komentar pada permintaan tarik untuk tumpukan baru (konsol)

Saat Anda membuat tumpukan, pada halaman Tentukan detail tumpukan, di bawah repositori definisi Template, pastikan sakelar Aktifkan komentar pada permintaan tarik diaktifkan. Ini adalah pengaturan default untuk tumpukan baru.

Untuk mengaktifkan komentar pada permintaan tarik untuk tumpukan (konsol) yang ada

1. Masuk ke AWS Management Console dan buka AWS CloudFormation konsol di <https://console.aws.amazon.com/cloudformation>.
2. Pada bilah navigasi di bagian atas layar, pilih tempat Wilayah AWS Anda membuat tumpukan.
3. Pada halaman Stacks, pilih tumpukan berjalan yang ingin Anda perbarui.
4. Pilih tab Git sync, lalu pilih Edit.
5. Pada halaman pengaturan sinkronisasi Edit Git, di bawah repositori definisi Template, aktifkan tombol Aktifkan komentar pada permintaan tarik.
6. Pilih Perbarui konfigurasi.

Dasbor status sinkronisasi Git

Untuk melihat status penerapan sinkronisasi AWS CloudFormation Git, pilih tumpukan di CloudFormation konsol dan pilih tab sinkronisasi Git.

Tab sinkronisasi Git dibagi menjadi dua panel, status sinkronisasi Git dan peristiwa sinkronisasi terbaru.

Status sinkronisasi Git

Panel atas menyediakan informasi berikut tentang konfigurasi sinkronisasi Git untuk tumpukan.

Repositori

Tautan ke repositori yang terhubung ke sinkronisasi Git

Penyedia repositori

Nama penyedia repositori

Cabang

Nama cabang yang disinkronkan Git sedang dipantau

Jalur file penyebaran

Jalur lengkap ke file penyebaran tumpukan untuk tumpukan ini

Status sinkronisasi repositori

Status operasi sinkronisasi terbaru

Pesan status sinkronisasi repositori

Pesan dari operasi sinkronisasi terbaru

Status sinkronisasi Git

Status sinkronisasi Git untuk tumpukan ini

Status penyediaan

Status operasi penyediaan

Di kanan atas panel, gunakan tombol berikut untuk memodifikasi atau memperbarui sinkronisasi Git:

- Edit - Edit konfigurasi sinkronisasi Git.
- Coba lagi komit terbaru - Perbarui tumpukan sesuai dengan komit terbaru ke repositori.
- Putuskan sambungan - Putuskan sambungan Git sync dari tumpukan.
- Segarkan - Segarkan kembali panel status sinkronisasi Git.

Acara sinkronisasi terbaru

Panel peristiwa sinkronisasi terbaru menampilkan tabel komit yang telah diterapkan ke tumpukan.

Anda dapat mengurutkan tabel menggunakan panah di header setiap kolom. Tabel dapat diurutkan dalam urutan naik atau turun sesuai dengan yang berikut:

- Tanggal
- ID Komit
- Peristiwa
- Tanggal
- Jenis Acara

Status tumpukan yang didukung

Sinkronisasi Git hanya dapat dikonfigurasi untuk tumpukan jika tumpukan berada di salah satu status yang didukung berikut:

- CREATE_COMPLETE
- UPDATE_COMPLETE
- UPDATE_ROLLBACK_COMPLETE
- IMPORT_COMPLETE
- IMPORT_ROLLBACK_COMPLETE

Tabel berikut berisi daftar lengkap kode status stack dengan deskripsi:

Status tumpukan dan status rinci opsional	Deskripsi
CREATE_COMPLETE	Pembuatan satu atau lebih tumpukan berhasil.
CREATE_IN_PROGRESS	Pembuatan satu atau lebih tumpukan yang sedang berlangsung.
CREATE_FAILED	Pembuatan satu atau lebih tumpukan gagal. Lihat peristiwa tumpukan untuk melihat pesan kesalahan yang terkait. Kemungkinan alasan pembuatan yang gagal termasuk izin yang tidak memadai untuk bekerja dengan semua sumber daya di tumpukan, nilai parameter yang ditolak oleh AWS layanan, atau batas waktu selama pembuatan sumber daya.
DELETE_COMPLETE	Penghapusan satu atau lebih tumpukan berhasil. Tumpukan yang dihapus dipertahankan dan dapat dilihat selama 90 hari.
DELETE_FAILED	Penghapusan satu atau lebih tumpukan gagal. Karena penghapusan gagal, Anda mungkin memiliki beberapa sumber daya yang masih berjalan; tetapi, Anda tidak dapat menggunakan atau memperbarui tumpukan tersebut. Hapus tumpukan lagi atau lihat peristiwa tumpukan untuk melihat pesan kesalahan yang terkait.

Status tumpukan dan status rinci opsional	Deskripsi
DELETE_IN_PROGRESS	Penghapusan satu atau lebih tumpukan yang sedang berlangsung.
REVIEW_IN_PROGRESS	Pembuatan satu atau lebih tumpukan yang sedang berlangsung dengan StackId yang diharapkan tetapi tanpa templat atau sumber daya apa pun.  Important Tumpukan dengan kode status ini dihitung terhadap jumlah tumpukan maksimum yang memungkinkan.
ROLLBACK_COMPLETE	Penghapusan satu atau beberapa tumpukan berhasil setelah pembuatan tumpukan gagal atau setelah pembuatan tumpukan dibatalkan secara eksplisit. Tumpukan kembali ke status kerja sebelumnya. Semua sumber daya yang dibuat selama operasi pembuatan tumpukan akan dihapus. Status ini hanya ada setelah pembuatan tumpukan gagal. Ini menandakan bahwa semua operasi dari tumpukan yang dibuat sebagian telah dibersihkan dengan baik. Ketika dalam status ini, hanya operasi penghapusan yang dapat dilakukan.
ROLLBACK_FAILED	Penghapusan satu atau beberapa tumpukan gagal setelah pembuatan tumpukan gagal atau setelah pembuatan tumpukan dibatalkan secara eksplisit. Hapus tumpukan atau lihat peristiwa tumpukan untuk melihat pesan kesalahan yang terkait.
ROLLBACK_IN_PROGRESS	Penghapusan satu atau beberapa tumpukan yang sedang berlangsung setelah pembuatan tumpukan yang gagal atau setelah pembuatan tumpukan yang dibatalkan secara eksplisit .
UPDATE_COMPLETE	Pembaruan satu atau lebih tumpukan berhasil.

Status tumpukan dan status rinci opsional	Deskripsi
UPDATE_COMPLETE_CLEANUP_IN_PROGRESS	Penghapusan sumber daya lama yang sedang berlangsung untuk satu atau beberapa tumpukan setelah pembaruan tumpukan berhasil. Untuk pembaruan tumpukan yang membutuhkan sumber daya untuk diganti, CloudFormation buat sumber daya baru terlebih dahulu dan kemudian hapus sumber daya lama untuk membantu mengurangi gangguan apa pun dengan tumpukan Anda. Dalam keadaan ini, tumpukan telah diperbarui dan dapat digunakan, tetapi CloudFormation masih menghapus sumber daya lama.
UPDATE_FAILED	Pembaruan satu atau lebih tumpukan gagal. Lihat peristiwa tumpukan untuk melihat pesan kesalahan yang terkait.
UPDATE_IN_PROGRESS	Pembaruan dari satu atau lebih tumpukan yang sedang berlangsung.
UPDATE_ROLLBACK_COMPLETE	Berhasil mengembalikan satu atau beberapa tumpukan ke status kerja sebelumnya setelah pembaruan tumpukan gagal.
UPDATE_ROLLBACK_COMPLETE_CLEANUP_IN_PROGRESS	Penghapusan sumber daya baru yang sedang berlangsung untuk satu atau lebih tumpukan setelah pembaruan tumpukan gagal. Dalam keadaan ini, tumpukan telah digulung kembali ke status kerja sebelumnya dan dapat digunakan, tetapi CloudFormation masih menghapus sumber daya baru yang dibuatnya selama pembaruan tumpukan.
UPDATE_ROLLBACK_FAILED	Gagal mengembalikan satu atau beberapa tumpukan ke status kerja sebelumnya setelah pembaruan tumpukan gagal. Saat dalam status ini, Anda dapat menghapus tumpukan atau melanjutkan rollback . Anda mungkin perlu memperbaiki kesalahan sebelum tumpukan Anda dapat kembali ke status kerja. Atau, Anda dapat menghubungi Dukungan untuk mengembalikan tumpukan ke status yang dapat digunakan.

Status tumpukan dan status rinci opsional	Deskripsi
UPDATE_ROLLBACK_IN_PROGRESS	Pengembalian yang sedang berlangsung dari satu atau lebih tumpukan ke status kerja sebelumnya setelah pembaruan tumpukan gagal.
IMPORT_IN_PROGRESS	Operasi impor sedang berlangsung.
IMPORT_COMPLETE	Operasi impor telah berhasil diselesaikan untuk semua sumber daya di tumpukan yang mendukung <code>resource import</code> .
IMPORT_ROLLBACK_IN_PROGRESS	Impor akan kembali ke konfigurasi templat sebelumnya.
IMPORT_ROLLBACK_FAILED	Operasi rollback impor gagal untuk setidaknya satu sumber daya di tumpukan. Hasil akan tersedia untuk sumber daya yang CloudFormation berhasil diimpor.
IMPORT_ROLLBACK_COMPLETE	Impor berhasil dikembalikan ke konfigurasi templat sebelumnya.

Mengelola ekstensi dengan CloudFormation registri

AWS CloudFormation Registri berfungsi sebagai hub terpusat untuk mengelola ekstensi yang dapat diintegrasikan ke dalam CloudFormation template di Akun AWS. Ekstensi mencakup jenis sumber daya, modul, dan Hooks dari AWS dan penerbit pihak ketiga, dan ekstensi kustom Anda sendiri. Registri membuatnya lebih mudah untuk menemukan dan menyediakan ekstensi di CloudFormation template Anda dengan cara yang sama seperti Anda menggunakan sumber daya AWS yang disediakan.

Bagian ini menjelaskan cara menggunakan CloudFormation registri untuk mengaktifkan ekstensi pihak ketiga di akun Anda, termasuk:

- Mengaktifkan ekstensi publik
- Mendaftarkan dan mengaktifkan ekstensi pribadi

Topik

- [Dokumentasi terkait](#)
- [CloudFormation konsep registri](#)
- [Lihat ekstensi yang tersedia dan diaktifkan di CloudFormation registri](#)
- [Gunakan ekstensi publik pihak ketiga dari CloudFormation registri](#)
- [Gunakan ekstensi pribadi pihak ketiga yang telah dibagikan dengan Anda](#)
- [Mengedit data konfigurasi untuk ekstensi di akun Anda](#)
- [Rekam jenis sumber daya di AWS Config](#)

Dokumentasi terkait

Jika Anda seorang pengembang yang tertarik untuk membuat ekstensi Anda sendiri, lihat dokumentasi berikut:

- [Mengembangkan modul menggunakan CloudFormation CLI di Panduan Pengguna](#) Antarmuka Baris CloudFormation Perintah
- [Membuat tipe sumber daya menggunakan CloudFormation CLI di Panduan Pengguna](#) Antarmuka Baris CloudFormation Perintah

- [Mengembangkan Hooks kustom menggunakan CloudFormation CLI](#) di AWS CloudFormation
Panduan Pengguna Hooks

Selain itu, semua jenis AWS sumber daya yang tersedia di CloudFormation registri dapat digunakan dengan AWS Cloud Control API, dengan atribut dan properti yang ditentukan dalam skema JSON standar. Untuk informasi selengkapnya, lihat [Panduan Pengguna Cloud Control API](#). Saat menggunakan Cloud Control API untuk melakukan operasi CRUDL (Buat, Baca, Perbarui, Hapus, Daftar) pada AWS sumber daya, Anda hanya dapat melakukannya pada AWS sumber daya Anda sendiri. Akun AWS

CloudFormation konsep registri

Topik ini menjelaskan konsep-konsep kunci untuk membantu Anda memahami dan mulai menggunakan CloudFormation registri.

Jenis ekstensi

CloudFormation Registri menawarkan jenis ekstensi berikut:

Kait

Hooks adalah pemeriksaan validasi yang memeriksa tumpukan atau sumber daya tertentu sebelum CloudFormation membuat, memperbarui, atau menghapusnya. Kait juga dapat dipanggil selama membuat operasi set perubahan. Mereka menyediakan mekanisme untuk menegakkan standar organisasi dan praktik terbaik dengan memvalidasi konfigurasi sumber daya terhadap persyaratan tertentu. Jika Hook mendeteksi konfigurasi apa pun yang tidak sesuai dengan logikanya, Hook dapat mengeluarkan peringatan atau gagal dalam proses penyediaan untuk mencegah sumber daya yang tidak sesuai diterapkan. Untuk informasi selengkapnya, lihat [Panduan Pengguna AWS CloudFormation Hooks](#).

Untuk dokumentasi khusus tentang Guard dan Lambda Hooks, lihat bagian Panduan Pengguna AWS CloudFormation Hooks berikut:

- [Kait Pelindung](#)
- [Kait Lambda](#)

Modul

Modul adalah konfigurasi sumber daya yang dapat digunakan kembali yang dapat disertakan di beberapa templat CloudFormation tumpukan. Mereka menyederhanakan pembuatan

dan pemeliharaan CloudFormation template dengan merangkum konfigurasi sumber daya yang kompleks atau sering digunakan menjadi komponen yang dapat digunakan kembali. Ini mendorong konsistensi dan standarisasi di seluruh penerapan infrastruktur organisasi Anda.

Jenis sumber daya

Jenis sumber daya memungkinkan Anda memodelkan dan mengotomatiskan sumber daya pihak ketiga atau sumber daya khusus yang tidak didukung secara native. CloudFormation Dengan mengembangkan jenis sumber daya, Anda dapat memperluas CloudFormation kemampuan untuk menyediakan dan mengelola sumber daya dari berbagai layanan pihak ketiga.

Ekstensi publik

Ekstensi publik adalah CloudFormation ekstensi yang diterbitkan secara publik di registri untuk digunakan oleh semua Akun AWS. Ini termasuk:

- **AWS ekstensi** — Ekstensi yang diterbitkan oleh selalu AWS bersifat publik, dan diaktifkan secara default, sehingga Anda tidak perlu mengambil tindakan apa pun sebelum menggunakannya di akun Anda. AWS mengontrol versi ekstensi ini, sehingga Anda selalu menggunakan versi terbaru yang tersedia.
- **Ekstensi pihak ketiga** — Ekstensi ini tersedia untuk penggunaan umum oleh penerbit selain. AWS Untuk menggunakan ekstensi publik pihak ketiga, Anda harus terlebih dahulu mengaktifkannya di akun dan Wilayah Anda.

Ekstensi yang diaktifkan

CloudFormation Registri di spesifik Anda Akun AWS mencakup tiga jenis ekstensi yang diaktifkan:

- **AWS ekstensi** - Semua ekstensi AWS publik diaktifkan secara otomatis.
- **Pihak ketiga yang diaktifkan** — Ini adalah salinan lokal dari ekstensi publik pihak ketiga yang telah Anda aktifkan secara eksplisit untuk akun dan Wilayah Anda. Saat Anda mengaktifkan ekstensi publik pihak ketiga, CloudFormation buat salinan lokal di registri akun Anda. Untuk informasi selengkapnya, lihat [Gunakan ekstensi publik pihak ketiga dari CloudFormation registri](#).
- **Terdaftar secara pribadi** — Ini adalah ekstensi pribadi yang tidak terdaftar di CloudFormation registri publik. Ini mungkin ekstensi yang Anda buat sendiri atau yang dibagikan dengan Anda oleh organisasi Anda atau pihak ketiga lainnya. Untuk menggunakan ekstensi pribadi semacam itu di akun Anda, Anda harus terlebih dahulu mendaftarkannya. Mendaftarkan ekstensi

mengunggah salinan ke CloudFormation registri di akun Anda dan mengaktifkannya. Untuk informasi selengkapnya, lihat [Gunakan ekstensi pribadi pihak ketiga yang telah dibagikan dengan Anda](#).

Menggunakan ekstensi yang terdaftar secara pribadi dan ekstensi publik yang diaktifkan dari penerbit pihak ketiga di akun Anda seperti menggunakannya di lingkungan kotak pasir. Ekstensi dikelola dengan kontrol versi, sehingga perilaku penyediaannya terkait dengan versi tertentu. Akibatnya, ekstensi ini berfungsi dengan cara yang sama seperti ekstensi publik, mengikuti aturan khusus versi yang sama.

Note

Ekstensi yang terdaftar secara pribadi dan ekstensi publik pihak ketiga yang diaktifkan dapat menerapkan event handler yang berjalan selama membuat, membaca, memperbarui, mencantumkan, dan menghapus operasi. Menggunakan ekstensi ini di CloudFormation tumpukan Anda dapat dikenakan biaya ke akun Anda, selain biaya apa pun untuk sumber daya yang dibuat. Untuk informasi lebih lanjut, lihat [Harga AWS CloudFormation](#).

Lihat ekstensi yang tersedia dan diaktifkan di CloudFormation registri

Untuk melihat ekstensi yang tersedia dan diaktifkan di CloudFormation registri, Anda dapat menggunakan AWS Management Console atau file AWS CLI.

Untuk melihat ekstensi yang tersedia dan diaktifkan (konsol)

1. Masuk ke AWS Management Console dan buka AWS CloudFormation konsol di <https://console.aws.amazon.com/cloudformation>.
2. Pada bilah navigasi di bagian atas layar, pilih Anda Wilayah AWS.
3. Dari panel navigasi, di bawah Registry, pilih kategori ekstensi yang ingin Anda lihat:
 - Ekstensi publik menampilkan ekstensi publik yang tersedia di akun Anda.
 1. Untuk jenis Filter, Ekstensi, pilih jenis ekstensi Anda: Jenis sumber daya, Modul, atau Kait.
 2. Untuk Filter, Penerbit, pilih penerbit: AWSatau pihak ketiga.

- Ekstensi yang diaktifkan menampilkan ekstensi publik dan pribadi yang diaktifkan di akun Anda.
 1. Pilih jenis ekstensi Anda: Jenis sumber daya, Modul, atau Kait.
 2. Gunakan menu drop-down Filter untuk lebih memilih ekstensi yang akan dilihat:
 - AWS— daftar ekstensi yang diterbitkan oleh AWS. Ekstensi yang diterbitkan oleh AWS diaktifkan secara default.
 - Pihak ketiga - mencantumkan ekstensi publik apa pun dari penerbit selain AWS yang telah Anda aktifkan di akun ini.
 - Terdaftar - daftar ekstensi pribadi apa pun yang telah Anda aktifkan di akun ini.
 3. Penerbit menampilkan ekstensi publik apa pun yang telah Anda publikasikan menggunakan akun ini.
 4. Cari atau pilih nama ekstensi untuk melihat detail ekstensi.

Untuk melihat ekstensi yang tersedia dan diaktifkan (AWS CLI)

Gunakan [list-types](#) perintah.

Gunakan ekstensi publik pihak ketiga dari CloudFormation registri

Untuk menggunakan ekstensi publik pihak ketiga di template Anda, Anda harus terlebih dahulu mengaktifkan ekstensi untuk akun dan Wilayah tempat Anda ingin menggunakannya. Mengaktifkan ekstensi membuatnya dapat digunakan dalam operasi tumpukan di akun dan Wilayah tempat ekstensi diaktifkan.

Saat Anda mengaktifkan ekstensi publik pihak ketiga, CloudFormation buat entri di registri ekstensi akun Anda untuk ekstensi yang diaktifkan sebagai ekstensi pribadi. Ini memungkinkan Anda untuk mengatur properti konfigurasi apa pun yang disertakan ekstensi. Properti konfigurasi menentukan bagaimana ekstensi dikonfigurasi untuk diberikan Akun AWS dan Wilayah.

Selain mengatur properti konfigurasi, Anda juga dapat menyesuaikan ekstensi dengan cara berikut:

- Tentukan peran eksekusi yang CloudFormation digunakan untuk mengaktifkan ekstensi, selain mengkonfigurasi logging untuk ekstensi.
- Tentukan apakah ekstensi diperbarui secara otomatis saat versi minor atau patch baru tersedia.
- Tentukan alias yang akan digunakan, bukan nama ekstensi publik pihak ketiga. Ini dapat membantu menghindari tabrakan penamaan antara ekstensi pihak ketiga.

Topik

- [Mengonfigurasi peran eksekusi dengan izin IAM dan kebijakan kepercayaan untuk akses ekstensi publik](#)
- [Secara otomatis menggunakan versi ekstensi baru](#)
- [Gunakan alias untuk merujuk ke ekstensi](#)
- [AWS CLI Perintah yang umum digunakan untuk bekerja dengan ekstensi publik](#)
- [Aktifkan ekstensi publik pihak ketiga di akun Anda](#)
- [Memperbarui ekstensi pihak ketiga publik di akun Anda](#)
- [Nonaktifkan ekstensi publik pihak ketiga di akun Anda](#)

Mengonfigurasi peran eksekusi dengan izin IAM dan kebijakan kepercayaan untuk akses ekstensi publik

Saat Anda mengaktifkan ekstensi publik dari CloudFormation registri, Anda dapat memberikan peran eksekusi yang memberikan izin CloudFormation yang diperlukan untuk memanggil ekstensi tersebut di wilayah Akun AWS dan wilayah Anda.

Izin yang diperlukan untuk peran eksekusi didefinisikan di bagian handler skema ekstensi. Anda harus membuat kebijakan IAM yang memberikan izin khusus yang diperlukan oleh ekstensi dan melampirkannya ke peran eksekusi.

Selain kebijakan izin, peran eksekusi juga harus memiliki kebijakan kepercayaan yang memungkinkan CloudFormation untuk mengambil peran. Ikuti panduan di [Membuat peran menggunakan kebijakan kepercayaan khusus](#) di Panduan Pengguna IAM untuk membuat peran dengan kebijakan kepercayaan khusus.

Hubungan kepercayaan

Berikut ini menunjukkan contoh kebijakan kepercayaan yang dapat Anda gunakan.

Anda dapat secara opsional membatasi ruang lingkup izin untuk pencegahan wakil kebingungan lintas layanan dengan menggunakan satu atau lebih kunci konteks kondisi global dengan bidang tersebut. Condition Untuk informasi selengkapnya, lihat [Pencegahan "confused deputy" lintas layanan](#).

- Tetapkan `aws:SourceAccount` nilainya ke ID akun Anda.
- Tetapkan `aws:SourceArn` nilainya ke ARN ekstensi Anda.

Contoh kebijakan kepercayaan 1

Berikut ini adalah contoh kebijakan kepercayaan peran IAM untuk ekstensi jenis sumber daya.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "resources.cloudformation.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        },
        "StringLike": {
          "aws:SourceArn": "arn:aws:cloudformation:us-west-2:123456789012:type/resource/Organization-Service-Resource/*"
        }
      }
    }
  ]
}
```

Contoh kebijakan kepercayaan 2

Berikut ini adalah contoh kebijakan kepercayaan peran IAM untuk ekstensi Hook.

```
{
  "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"
    },
    "StringLike":{
      "aws:SourceArn":"arn:aws:cloudformation:us-
west-2:123456789012:type/hook/Organization-Service-Hook/*"
    }
  }
}
```

Secara otomatis menggunakan versi ekstensi baru

Saat Anda mengaktifkan ekstensi, Anda juga dapat menentukan jenis ekstensi untuk menggunakan versi minor terbaru. Jenis ekstensi Anda memperbarui versi minor, setiap kali penerbit merilis versi baru pada ekstensi Anda yang diaktifkan.

Misalnya, saat berikutnya Anda melakukan operasi tumpukan, seperti membuat atau memperbarui tumpukan, menggunakan templat yang menyertakan ekstensi itu, CloudFormation menggunakan versi minor yang baru.

Memperbarui ke versi ekstensi baru, baik secara otomatis maupun manual dan tidak memengaruhi instance ekstensi apa pun yang sudah disediakan di tumpukan.

CloudFormation memperlakukan pembaruan versi utama ekstensi sebagai berpotensi mengandung perubahan yang melanggar, sehingga mengharuskan Anda memperbarui secara manual ke versi utama ekstensi yang baru.

Ekstensi AWS yang diterbitkan oleh diaktifkan secara default untuk semua akun dan Wilayah yang tersedia, dan selalu gunakan versi terbaru yang tersedia di masing-masing akun Wilayah AWS.

Important

Karena Anda mengontrol jika dan kapan ekstensi diperbarui ke versi terbaru di akun Anda, Anda bisa mendapatkan versi berbeda dari ekstensi yang sama yang digunakan di akun dan Wilayah yang berbeda.

Ini berpotensi menyebabkan hasil yang tidak terduga saat menggunakan templat yang sama, yang berisi ekstensi itu, di seluruh akun dan Wilayah tersebut.

Gunakan alias untuk merujuk ke ekstensi

Anda tidak dapat mengaktifkan lebih dari satu ekstensi dengan nama tertentu di wilayah tertentu Akun AWS. Karena penerbit yang berbeda mungkin menawarkan ekstensi publik dengan nama ekstensi yang sama, CloudFormation Anda dapat menentukan alias untuk ekstensi publik pihak ketiga yang Anda aktifkan.

Jika Anda menentukan alias untuk ekstensi, CloudFormation memperlakukan alias sebagai nama jenis ekstensi dalam akun dan Wilayah. Anda harus menggunakan alias untuk merujuk ke ekstensi di template, panggilan API, dan CloudFormation konsol Anda.

Alias ekstensi harus unik dalam akun dan Wilayah tertentu. Anda dapat mengaktifkan sumber daya publik yang sama beberapa kali di akun dan Wilayah yang sama, menggunakan alias nama tipe yang berbeda.

Important

Meskipun alias ekstensi hanya diperlukan untuk menjadi unik di akun dan Wilayah tertentu, kami sangat menyarankan agar pengguna tidak menetapkan alias yang sama ke ekstensi publik pihak ketiga yang berbeda di seluruh akun dan Wilayah. Melakukannya dapat menyebabkan hasil yang tidak terduga saat menggunakan templat yang berisi alias ekstensi di beberapa akun atau Wilayah.

AWS CLI Perintah yang umum digunakan untuk bekerja dengan ekstensi publik

Perintah yang umum digunakan untuk bekerja dengan ekstensi publik meliputi:

- [activate-type](#) untuk mengaktifkan modul pihak ketiga publik atau jenis sumber daya di akun Anda.
- [set-type-configuration](#) untuk menentukan data konfigurasi untuk ekstensi di akun Anda dan untuk menonaktifkan dan mengaktifkan Hooks.
- [list-types](#) untuk membuat daftar ekstensi di akun Anda.
- [describe-type](#) untuk mengembalikan informasi terperinci tentang ekstensi tertentu atau versi ekstensi tertentu, termasuk data konfigurasi saat ini.
- [set-type-default-version](#) untuk menentukan versi ekstensi mana yang merupakan versi default.

- [deactivate-type](#) untuk menonaktifkan modul pihak ketiga publik atau jenis sumber daya yang sebelumnya diaktifkan di akun Anda.

Aktifkan ekstensi publik pihak ketiga di akun Anda

Topik berikut menunjukkan kepada Anda cara mengaktifkan ekstensi publik pihak ketiga di akun Anda, yang membuatnya dapat digunakan di akun dan Wilayah tempat ia diaktifkan.

Note

Sebelum melanjutkan, konfirmasikan bahwa Anda telah membuat [Peran IAM](#) yang akan Anda gunakan dengan ekstensi ini.

Topik

- [Aktifkan ekstensi publik \(konsol\)](#)
- [Aktifkan ekstensi publik \(AWS CLI\)](#)

Aktifkan ekstensi publik (konsol)

Ikuti langkah-langkah di bagian ini untuk menggunakan konsol untuk:

- Aktifkan ekstensi publik pihak ketiga
- Tentukan data konfigurasi ekstensi tambahan untuk akun Anda

Untuk mengaktifkan ekstensi publik untuk digunakan di akun Anda

1. Masuk ke AWS Management Console dan buka AWS CloudFormation konsol di <https://console.aws.amazon.com/cloudformation>.
2. Pada bilah navigasi di bagian atas layar, pilih Anda Wilayah AWS.
3. Dari panel navigasi, di bawah Registry, pilih Ekstensi publik.
4. Gunakan Filter untuk memilih jenis ekstensi, dan pilih Pihak ketiga. (Ekstensi yang diterbitkan oleh AWS diaktifkan secara default.)
5. Pilih ekstensi, lalu pilih Aktifkan.

Jika beberapa versi ekstensi tersedia, Anda dapat menggunakan menu Versi untuk memilih versi ekstensi yang ingin Anda aktifkan. Default adalah versi terbaru.

6. Untuk nama Ekstensi, Anda dapat tetap memilih Use default, atau memilih Override default, lalu masukkan alias jenis ekstensi yang ingin Anda gunakan dengan ekstensi ini. Alias harus mengikuti format yang disarankan untuk jenis ekstensi. Untuk informasi selengkapnya, lihat [Gunakan alias untuk merujuk ke ekstensi](#).
7. Jika ekstensi yang Anda aktifkan adalah tipe Hook atau sumber daya, untuk peran Eksekusi ARN, tentukan peran IAM CloudFormation untuk diasumsikan saat memanggil ekstensi. Untuk informasi selengkapnya, lihat [Mengonfigurasi peran eksekusi dengan izin IAM dan kebijakan kepercayaan untuk akses ekstensi publik](#).
8. Untuk konfigurasi Logging, tentukan informasi konfigurasi logging untuk ekstensi, jika diinginkan. Misalnya:

```
{
  "logRoleArn": "arn:aws:iam::account:role/rolename",
  "logGroupName": "log-group-name"
}
```

Informasi konfigurasi logging tidak diperlukan tetapi disarankan untuk tujuan debugging. Untuk menggunakan konfigurasi logging dengan Hooks, tambahkan kebijakan kepercayaan yang sama dengan peran eksekusi yang ditentukan, sehingga peran log dapat menulis log ke grup log Anda.

logRoleArndan nama logGroupName kunci peka huruf besar/kecil.

9. Untuk Pembuatan Versi, Pembaruan otomatis, pilih cara menerima pembaruan.
 - Aktif - Secara otomatis memperbarui ke versi minor terbaru. Versi utama diperbarui secara manual.
 - Mati - Jangan pernah memperbarui ke versi terbaru secara otomatis. Semua versi diperbarui secara manual.

Untuk informasi selengkapnya, lihat [Secara otomatis menggunakan versi ekstensi baru](#).

Jika ekstensi memerlukan konfigurasi tambahan, Anda memiliki opsi untuk menentukan data konfigurasi sekarang, atau setelah ekstensi diaktifkan.

 Important

Jika ekstensi yang Anda aktifkan adalah Hook, langkah ini diperlukan. Anda harus menentukan `ENABLED` untuk `HookInvocationStatus` properti. Operasi ini memungkinkan properti Hook yang didefinisikan di `properties` bagian skema Hook. Untuk informasi selengkapnya, lihat [Referensi sintaks skema konfigurasi Hook](#) di Panduan Pengguna AWS CloudFormation Hooks.

Untuk menentukan data konfigurasi

1. Untuk Konfigurasi, pilih Konfigurasi sekarang, lalu pilih Aktifkan ekstensi.

CloudFormation menampilkan halaman ekstensi Konfigurasi. Untuk melihat skema konfigurasi saat ini untuk ekstensi, pastikan Lihat skema konfigurasi sudah diaktifkan.

2. Dalam kotak teks Konfigurasi JSON, masukkan string JSON yang mewakili data konfigurasi yang ingin Anda tentukan untuk ekstensi ini. JSON yang Anda tentukan harus memvalidasi terhadap skema konfigurasi ekstensi.
3. Pilih Konfigurasi ekstensi.

Jika Anda lebih suka mengonfigurasi ekstensi setelah aktivasi, Anda dapat melewati langkah ini dan memberikan data konfigurasi di lain waktu.

1. Untuk Konfigurasi, pilih Konfigurasi nanti, lalu pilih Aktifkan ekstensi.
2. Setelah ekstensi diaktifkan, Anda dapat mengonfigurasinya dengan menavigasi ke ekstensi dari halaman ekstensi yang diaktifkan dan memberikan data konfigurasi.

Aktifkan ekstensi publik (AWS CLI)

Ikuti langkah-langkah di bagian ini untuk menggunakan AWS CLI to:

- Aktifkan ekstensi publik pihak ketiga
- Tentukan data konfigurasi ekstensi tambahan untuk akun Anda

Aktifkan Hooks publik

Dengan mengaktifkan Hooks di akun Anda, Anda mengotorisasi Hook untuk menggunakan izin yang ditentukan dari akun Anda. Akun AWS CloudFormation menghapus izin yang tidak diperlukan sebelum meneruskan izin Anda ke Hook. CloudFormation merekomendasikan pelanggan atau pengguna Hook untuk meninjau izin Hook dan mengetahui izin apa yang diizinkan oleh Hooks sebelum mengaktifkan Hooks di akun Anda.

Untuk mengaktifkan Hook publik untuk digunakan di akun Anda (AWS CLI)

1. Dapatkan ARN untuk Hook Anda dan simpan. Anda bisa mendapatkan ARN dari Hook menggunakan or. AWS Management Console AWS CLI Untuk mengetahui informasi selengkapnya, lihat [Lihat ekstensi yang tersedia dan diaktifkan di CloudFormation registri](#).

```
export HOOK_TYPE_ARN="arn:aws:cloudformation:us-west-2:123456789012:type/hook/  
Organization-Service-Hook/"
```

2. Gunakan [set-type-configuration](#) perintah untuk menentukan data konfigurasi. JSON yang Anda lewati --configuration harus memvalidasi terhadap skema konfigurasi Hook. Untuk mengaktifkan Hook untuk semua operasi tumpukan, Anda harus mengatur HookInvocationStatus properti ENABLED di HookConfiguration bagian.

```
aws cloudformation set-type-configuration \  
  --configuration '{"CloudFormationConfiguration":{"HookConfiguration":  
  {"HookInvocationStatus": "ENABLED", "FailureMode": "FAIL", "Properties":{}}}}' \  
  --type-arn $HOOK_TYPE_ARN --region us-west-2
```

Untuk informasi selengkapnya tentang opsi HookConfiguration konfigurasi, lihat [Referensi sintaks skema konfigurasi Hook di Panduan](#) Pengguna AWS CloudFormation Hooks.

Aktifkan modul publik dan jenis sumber daya

Untuk mengaktifkan ekstensi publik untuk digunakan di akun Anda (AWS CLI)

- Gunakan [activate-type](#) perintah untuk mengaktifkan ekstensi, dan menentukan apakah akan auto update ekstensi setiap kali versi minor baru dari ekstensi diterbitkan.

Contoh di bawah ini menentukan Nama Sumber Daya Amazon publik (ARN) dari ekstensi publik untuk mengaktifkan akun ini. Selain itu, ini menentukan bahwa CloudFormation memperbarui ekstensi setiap kali versi minor baru diterbitkan.

```
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
```

Perintah ini mengembalikan ARN dari ekstensi yang diaktifkan.

```
{  
  "Arn": "arn:aws:cloudformation:us-west-2:123456789012:type/resource/My-  
Resource-Example"  
}
```

Memperbarui versi ekstensi publik (AWS CLI)

Gunakan [activate-type](#) untuk mengaktifkan ekstensi lagi.

Gunakan `--version-bump` opsi untuk menentukan apakah akan memperbarui ekstensi ke MAJOR versi terbaru atau MINOR versi terbaru.

```
aws cloudformation activate-type --type RESOURCE \  
  --type-name Example::Test::1234567890abcdef0 \  
  --type-name-alias Example::Test::Alias \  
  --version-bump MAJOR --region us-west-2
```

Memperbarui ekstensi pihak ketiga publik di akun Anda

Setelah mengaktifkan ekstensi publik pihak ketiga, Anda dapat memperbarui sebagian besar detail ekstensi dari akun Anda.

Untuk memperbarui ekstensi publik di akun Anda (konsol)

1. Masuk ke AWS Management Console dan buka AWS CloudFormation konsol di <https://console.aws.amazon.com/cloudformation>.
2. Pada bilah navigasi di bagian atas layar, pilih Anda Wilayah AWS.

3. Dari panel navigasi, di bawah Registry, pilih Ekstensi yang diaktifkan.
4. Temukan ekstensi yang ingin Anda perbarui dan pilih. Untuk informasi selengkapnya, lihat [Lihat ekstensi yang tersedia dan diaktifkan di CloudFormation registri](#).
5. Dari menu Tindakan, pilih Edit, lalu opsi pengeditan yang sesuai:
 - Untuk memperbarui skema konfigurasi, lihat [Mengedit data konfigurasi untuk ekstensi di akun Anda](#).
 - Untuk mengaktifkan atau menonaktifkan pembaruan otomatis:
 1. Pilih Edit pembaruan otomatis.
 2. Pilih Aktif atau Mati, lalu pilih Simpan. Untuk informasi selengkapnya, lihat [Secara otomatis menggunakan versi ekstensi baru](#).
 - Untuk memperbarui peran eksekusi:
 1. Pilih Edit peran eksekusi.
 2. Tentukan ARN peran IAM yang CloudFormation ingin Anda gunakan saat menjalankan ekstensi ini, lalu pilih Simpan. Untuk informasi selengkapnya, lihat [Mengonfigurasi peran eksekusi dengan izin IAM dan kebijakan kepercayaan untuk akses ekstensi publik](#).
 - Untuk memperbarui konfigurasi pencatatan:
 1. Pilih Edit konfigurasi logging.
 2. Edit konfigurasi logging JSON, lalu pilih Simpan.

Nonaktifkan ekstensi publik pihak ketiga di akun Anda

Jika Anda tidak lagi memerlukan ekstensi publik pihak ketiga yang diaktifkan, gunakan prosedur berikut untuk menonaktifkannya di akun Anda.

Topik

- [Nonaktifkan ekstensi publik di akun Anda \(konsol\)](#)
- [Nonaktifkan ekstensi publik di akun Anda \(AWS CLI\)](#)
- [Nonaktifkan Hook di akun Anda \(AWS CLI\)](#)

Nonaktifkan ekstensi publik di akun Anda (konsol)

Untuk menonaktifkan ekstensi publik di akun Anda

1. Masuk ke AWS Management Console dan buka AWS CloudFormation konsol di <https://console.aws.amazon.com/cloudformation>.
2. Pada bilah navigasi di bagian atas layar, pilih Anda Wilayah AWS.
3. Dari panel navigasi, di bawah Registry, pilih Ekstensi yang diaktifkan.
4. Temukan ekstensi yang ingin Anda nonaktifkan dan pilih. Untuk informasi selengkapnya, lihat [Lihat ekstensi yang tersedia dan diaktifkan di CloudFormation registri](#).
5. Dari menu Tindakan, pilih Nonaktifkan.
6. Pilih Nonaktifkan.

Nonaktifkan ekstensi publik di akun Anda (AWS CLI)

Gunakan yang berikut ini [deactivate-type](#) perintah.

```
aws cloudformation deactivate-type --type MODULE \  
  --type-name Example::Test::Type::MODULE \  
  --region us-west-2
```

Nonaktifkan Hook di akun Anda (AWS CLI)

Menonaktifkan Hook mencegah Hook berjalan di Akun AWS tanpa melepaskannya.

Gunakan [set-type-configuration](#) perintah dan tentukan HookInvocationStatus DISABLED untuk menonaktifkan Hook.

Contoh berikut menentukan Wilayah AWS dan Amazon Resource Name (ARN) dari Hook yang sedang dinonaktifkan.

```
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
```

Untuk informasi selengkapnya, lihat [Menonaktifkan dan mengaktifkan AWS CloudFormation Hooks](#) di Panduan Pengguna AWS CloudFormation Hooks.

Gunakan ekstensi pribadi pihak ketiga yang telah dibagikan dengan Anda

Untuk menggunakan ekstensi pribadi pihak ketiga yang telah dibagikan dengan Anda, Anda harus terlebih dahulu mendaftarkannya CloudFormation, di akun dan Wilayah tempat Anda ingin menggunakannya. Mendaftarkan ekstensi mengunggah salinannya ke CloudFormation registri di akun Anda, dan mengaktifkannya. Setelah Anda mendaftarkan ekstensi pribadi, itu akan muncul di CloudFormation registri untuk itu Akun AWS dan Wilayah, dan Anda dapat menggunakannya di template tumpukan Anda.

Topik

- [Izin IAM untuk mendaftarkan ekstensi pribadi pihak ketiga](#)
- [AWS CLI Perintah yang umum digunakan untuk bekerja dengan ekstensi pribadi](#)
- [Daftarkan ekstensi pribadi pihak ketiga di akun Anda](#)
- [Membatalkan pendaftaran ekstensi pribadi pihak ketiga yang sebelumnya Anda daftarkan di akun Anda](#)

Izin IAM untuk mendaftarkan ekstensi pribadi pihak ketiga

Sebagai bagian dari mendaftarkan ekstensi pribadi, Anda dapat menentukan bucket Amazon S3 yang berisi paket proyek ekstensi. Paket ini berisi file sumber yang diperlukan untuk ekstensi yang ingin Anda daftarkan. Pengguna yang mendaftarkan ekstensi harus dapat mengakses paket proyek di bucket Amazon S3 itu. Untuk melakukannya, pengguna harus [GetObject](#) izin untuk paket ekstensi.

Ini benar apakah Anda menggunakan [register-type](#) perintah dari AWS CLI, atau [submit](#) perintah CloudFormation CLI.

Untuk informasi selengkapnya, lihat [Tindakan, Sumber Daya, dan Kunci Kondisi untuk Amazon S3](#) di Referensi Otorisasi Layanan.

AWS CLI Perintah yang umum digunakan untuk bekerja dengan ekstensi pribadi

Perintah yang umum digunakan untuk bekerja dengan ekstensi pribadi meliputi:

- [register-type](#) untuk mendaftarkan ekstensi pribadi di akun Anda.
- [describe-type-registration](#) untuk mengembalikan status permintaan pendaftaran saat ini.

- [list-types](#) untuk membuat daftar ekstensi di akun Anda.
- [describe-type](#) untuk mengembalikan informasi terperinci tentang ekstensi tertentu atau versi ekstensi tertentu, termasuk data konfigurasi saat ini.
- [set-type-configuration](#) untuk menentukan data konfigurasi untuk ekstensi di akun Anda dan untuk menonaktifkan dan mengaktifkan Hooks.
- [set-type-default-version](#) untuk menentukan versi ekstensi mana yang merupakan versi default.
- [deregister-type](#) untuk menghapus ekstensi pribadi atau versi ekstensi dari akun Anda.

Daftarkan ekstensi pribadi pihak ketiga di akun Anda

Topik ini mencakup langkah-langkah untuk mendaftarkan ekstensi pribadi pihak ketiga yang dibagikan kepada Anda sehingga tersedia untuk digunakan di akun Anda.

Note

Sebelum melanjutkan, konfirmasikan bahwa Anda memiliki persyaratan [Izin IAM](#) untuk mendaftarkan ekstensi pribadi.

Untuk mendaftarkan ekstensi pribadi yang dibagikan dengan Anda (AWS CLI)

1. Temukan bucket Amazon S3 yang berisi paket proyek untuk ekstensi pribadi yang ingin Anda daftarkan di akun Anda.
2. Gunakan [register-type](#) perintah untuk mendaftarkan ekstensi pribadi di akun Anda.

Misalnya, perintah berikut mendaftarkan jenis `My::Resource::Example` sumber daya dalam yang ditentukan Akun AWS.

```
aws cloudformation register-type --type RESOURCE \  
  --type-name My::Resource::Example \  
  --schema-handler-package [s3 object path] --region us-west-2
```

`RegisterType` adalah operasi asinkron, dan mengembalikan token pendaftaran yang dapat Anda gunakan untuk melacak kemajuan permintaan pendaftaran Anda.

```
{  
  "RegistrationToken": "f5525280-104e-4d35-bef5-8f1fexample"
```

```
}
```

Jika ekstensi Anda memanggil AWS APIs sebagai bagian dari fungsinya, Anda harus membuat peran eksekusi IAM yang mencakup izin yang diperlukan untuk memanggilnya AWS APIs, dan menyediakan peran eksekusi tersebut di akun Anda. Anda kemudian dapat menentukan peran eksekusi ini menggunakan `--execution-role-arn` opsi. CloudFormation kemudian mengasumsikan bahwa peran eksekusi untuk menyediakan jenis sumber daya Anda dengan kredensi yang sesuai.

```
--execution-role-arn arn:aws:iam::123456789012:role/MyIAMRole
```

3. (Opsional) Gunakan token pendaftaran dengan [describe-type-registration](#) perintah untuk melacak kemajuan permintaan pendaftaran Anda.

Ketika CloudFormation menyelesaikan permintaan pendaftaran, itu menetapkan status kemajuan permintaan keCOMPLETE.

Contoh berikut menggunakan token registrasi yang dikembalikan oleh `describe-type-registration` perintah di atas untuk mengembalikan informasi status pendaftaran.

```
aws cloudformation describe-type-registration \  
  --registration-token f5525280-104e-4d35-bef5-8f1fexample \  
  --region us-west-2
```

Perintah mengembalikan output berikut.

```
{  
  "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

Jika ekstensi yang Anda daftarkan adalah Hook, langkah selanjutnya ini diperlukan. Anda harus menentukan ENABLED untuk HookInvocationStatus properti. Operasi

ini memungkinkan properti Hook yang didefinisikan di `properties` bagian skema Hook. Untuk informasi selengkapnya, lihat [Referensi sintaks skema konfigurasi Hook](#) di Panduan Pengguna AWS CloudFormation Hooks.

Untuk menentukan data konfigurasi untuk Hook (AWS CLI)

1. Dapatkan ARN untuk Hook Anda dan simpan. Anda bisa mendapatkan ARN dari Hook menggunakan `aws` atau AWS Management Console. Untuk informasi selengkapnya, lihat [Lihat ekstensi yang tersedia dan diaktifkan di CloudFormation registri](#).

```
export HOOK_TYPE_ARN="arn:aws:cloudformation:us-west-2:123456789012:type/hook/  
Organization-Service-Hook/"
```

2. Gunakan [set-type-configuration](#) perintah untuk menentukan data konfigurasi. JSON yang Anda lewati `--configuration` harus memvalidasi terhadap skema konfigurasi Hook. Untuk mengaktifkan Hook, Anda harus mengatur `HookInvocationStatus` properti `ENABLED` di `HookConfiguration` bagian.

```
aws cloudformation set-type-configuration \  
  --configuration '{"CloudFormationConfiguration":{"HookConfiguration":  
  {"HookInvocationStatus": "ENABLED", "FailureMode": "FAIL", "Properties":{}}}}' \  
  --type-arn $HOOK_TYPE_ARN --region us-west-2
```

Untuk informasi selengkapnya, lihat [Referensi sintaks skema konfigurasi Hook](#) di Panduan Pengguna AWS CloudFormation Hooks.

Membatalkan pendaftaran ekstensi pribadi pihak ketiga yang sebelumnya Anda daftarkan di akun Anda

Untuk menghapus ekstensi pribadi pihak ketiga atau versi ekstensi, gunakan [deregister-type](#) perintah.

Anda dapat membatalkan pendaftaran versi ekstensi tertentu, atau ekstensi secara keseluruhan. Untuk membatalkan pendaftaran ekstensi, Anda harus membatalkan pendaftaran secara individual semua versi ekstensi yang terdaftar. Jika ekstensi hanya memiliki satu versi terdaftar, membatalkan pendaftaran versi tersebut mengakibatkan ekstensi itu sendiri dideregistrasi. Anda tidak dapat membatalkan pendaftaran versi default ekstensi, kecuali itu satu-satunya versi terdaftar dari ekstensi itu, dalam hal ini ekstensi itu sendiri juga dideregistrasi.

Contoh berikut membatalkan pendaftaran ekstensi. Ganti nilai sampel untuk `--type` dan `--type-name`.

```
aws cloudformation deregister-type --type MODULE \  
  --type-name My::S3::SampleBucket::MODULE \  
  --region us-west-2
```

Untuk membatalkan pendaftaran versi tertentu dari ekstensi Anda, tentukan `--version-id` opsi dalam perintah.

Mengedit data konfigurasi untuk ekstensi di akun Anda

Topik ini memberikan panduan tentang mengedit data konfigurasi untuk ekstensi di akun Anda dalam Wilayah tertentu. Ekstensi dapat menyertakan properti konfigurasi yang berlaku untuk semua instance ekstensi untuk akun dan Wilayah tertentu. Ini didefinisikan oleh pembuat ekstensi dalam definisi konfigurasi ekstensi. Jika ada properti yang diperlukan dalam definisi konfigurasi ekstensi, Anda harus menentukan properti tersebut sebelum dapat menggunakan ekstensi di akun dan Wilayah Anda.

Untuk informasi selengkapnya tentang bagaimana definisi konfigurasi didefinisikan saat mengembangkan ekstensi, lihat dokumentasi berikut.

- [Referensi sintaks skema konfigurasi hook](#)
- [Mendefinisikan konfigurasi tingkat akun ekstensi](#)

Topik

- [Izin diperlukan untuk menggunakan referensi dinamis](#)
- [Mengedit data konfigurasi untuk ekstensi \(konsol\)](#)
- [Mengedit data konfigurasi untuk ekstensi \(AWS CLI\)](#)

Izin diperlukan untuk menggunakan referensi dinamis

Jika data konfigurasi Anda menyertakan referensi dinamis ke nilai yang disimpan dalam AWS Systems Manager atau AWS Secrets Manager, peran apa pun yang digunakan untuk menyediakan tipe (misalnya, saat membuat atau memperbarui tumpukan) harus memiliki izin yang tepat untuk mengambil nilai tersebut. Secara khusus:

- Jika data konfigurasi berisi parameter yang disimpan di AWS Systems Manager Parameter Store, pengguna atau peran yang digunakan untuk menyediakan tipe harus memiliki izin untuk memanggil [GetParameter](#).
- Jika data konfigurasi berisi rahasia yang disimpan AWS Secrets Manager, pengguna atau peran yang digunakan untuk menyediakan tipe harus memiliki izin untuk memanggil [GetSecretValue](#).

Untuk informasi selengkapnya, lihat [Dapatkan nilai yang disimpan di layanan lain menggunakan referensi dinamis](#).

Mengedit data konfigurasi untuk ekstensi (konsol)

Ikuti langkah-langkah di bagian ini untuk menggunakan konsol untuk:

- Melihat data konfigurasi saat ini untuk ekstensi
- Perbarui data konfigurasi ekstensi untuk akun Anda

Untuk melihat data konfigurasi saat ini untuk ekstensi

1. Masuk ke AWS Management Console dan buka AWS CloudFormation konsol di <https://console.aws.amazon.com/cloudformation>.
2. Pada bilah navigasi di bagian atas layar, pilih Anda Wilayah AWS.
3. Dari panel navigasi, di bawah Registry, pilih Ekstensi yang diaktifkan.
4. Temukan ekstensi yang ingin Anda lihat. Untuk informasi selengkapnya, lihat [Lihat ekstensi yang tersedia dan diaktifkan di CloudFormation registri](#).
5. Pilih ekstensi untuk melihat detail ekstensi.
6. Pada halaman detail ekstensi, pilih tab Konfigurasi.
7. Perluas tab skema Konfigurasi untuk melihat skema konfigurasi yang ditentukan untuk ekstensi.
8. Perluas tab Konfigurasi untuk melihat konfigurasi saat ini yang telah Anda tetapkan untuk ekstensi ini.

Untuk memperbarui data konfigurasi untuk ekstensi

1. Pada halaman detail ekstensi, dari tab Konfigurasi, pilih Edit konfigurasi.

Atau, dari Tindakan, pilih Edit, lalu pilih Edit konfigurasi.

CloudFormation menampilkan halaman ekstensi Konfigurasi. Pastikan skema konfigurasi View diaktifkan untuk melihat skema definisi konfigurasi ekstensi saat ini.

2. Di kotak teks Konfigurasi JSON, masukkan string JSON yang mewakili skema konfigurasi yang ingin Anda atur untuk ekstensi ini. Itu harus memvalidasi terhadap skema yang didefinisikan dalam skema Konfigurasi.
3. Pilih Konfigurasi ekstensi.

Mengedit data konfigurasi untuk ekstensi (AWS CLI)

Ikuti langkah-langkah di bagian ini untuk menggunakan AWS CLI to:

- Melihat data konfigurasi saat ini untuk ekstensi
- Perbarui data konfigurasi ekstensi untuk akun Anda

Untuk melihat data konfigurasi saat ini untuk ekstensi

- Gunakan [describe-type](#) perintah untuk mengembalikan informasi rinci tentang ekstensi. ConfigurationSchemaElemen output berisi definisi konfigurasi ekstensi saat ini di Wilayah tertentu.

Atau, gunakan [batch-describe-type-configurations](#) perintah untuk mengembalikan data konfigurasi tentang beberapa ekstensi.

Untuk memperbarui data konfigurasi untuk ekstensi

- Gunakan [set-type-configuration](#) perintah untuk menentukan data konfigurasi. JSON yang Anda lewati --configuration harus memvalidasi terhadap skema konfigurasi ekstensi.

Dalam contoh berikut, set-type-configuration perintah menentukan data konfigurasi `{"CredentialKey": "testUserCredential"}` untuk --configuration opsi.

```
aws cloudformation set-type-configuration --type RESOURCE \
  --type-name My::Resource::Example \
  --configuration-alias default \
  --configuration {"CredentialKey": "testUserCredential"} \
  --region us-west-2
```

Rekam jenis sumber daya di AWS Config

Anda dapat menentukan yang AWS Config secara otomatis melacak jenis sumber daya pribadi Anda dan mencatat perubahan pada sumber daya tersebut sebagai item konfigurasi. Ini memungkinkan Anda untuk melihat riwayat konfigurasi untuk jenis sumber daya pribadi ini, selain menulis Aturan AWS Config aturan untuk memverifikasi praktik terbaik konfigurasi. AWS Config diperlukan untuk ekstensi Hook.

Untuk secara AWS Config otomatis melacak jenis sumber daya pribadi Anda:

- Kelola sumber daya melalui CloudFormation. Ini termasuk melakukan semua operasi pembuatan, pembaruan, dan penghapusan sumber daya melalui CloudFormation.

Note

Jika Anda menggunakan peran IAM untuk melakukan operasi tumpukan, peran IAM tersebut harus memiliki izin untuk memanggil tindakan berikut: AWS Config

- [PutResourceConfig](#)
- [DeleteResourceConfig](#)

- Konfigurasi AWS Config untuk merekam semua jenis sumber daya. Untuk informasi selengkapnya, lihat [Merekam Konfigurasi untuk Sumber Daya Pihak Ketiga menggunakan Panduan AWS Config Pengembang](#). AWS CLI

Note

AWS Config tidak mendukung perekaman sumber daya pribadi yang berisi properti yang didefinisikan sebagai wajib dan hanya ditulis.

Secara desain, properti sumber daya yang didefinisikan sebagai write-only tidak dikembalikan dalam skema yang digunakan untuk membuat AWS Config item konfigurasi. Karena itu, termasuk properti yang didefinisikan sebagai write-only dan required akan menyebabkan pembuatan item konfigurasi gagal, karena properti wajib tidak akan ada.

Untuk melihat skema yang akan digunakan untuk membuat item konfigurasi, Anda dapat meninjau schema properti [DescribeTypeTindakan](#).

Untuk informasi selengkapnya tentang item konfigurasi, lihat [Item konfigurasi](#) di Panduan AWS Config Pengembang.

Mencegah properti sensitif dicatat dalam item konfigurasi

Jenis sumber daya Anda mungkin berisi properti yang Anda anggap sebagai informasi sensitif, seperti kata sandi, rahasia, atau data sensitif lainnya, yang tidak ingin Anda rekam sebagai bagian dari item konfigurasi. Untuk mencegah properti dicatat dalam item konfigurasi, Anda bisa menyertakan properti itu dalam daftar `writeOnlyProperties` di skema tipe sumber daya Anda. Properti sumber daya yang terdaftar sebagai `writeOnlyProperties` dapat ditentukan oleh pengguna, tetapi tidak akan dikembalikan oleh `list` permintaan `read` atau.

Untuk informasi selengkapnya, silakan lihat [writeOnlyProperties](#) di Panduan Pengguna CloudFormation CLI.

Pengiriman terus menerus dengan CodePipeline

Pengiriman berkelanjutan adalah praktik rilis di mana perubahan kode secara otomatis dibuat, diuji, dan disiapkan untuk rilis ke produksi. Dengan CloudFormation dan CodePipeline, Anda dapat menggunakan pengiriman berkelanjutan untuk secara otomatis membuat dan menguji perubahan pada CloudFormation templat Anda sebelum mempromosikannya ke tumpukan produksi. Proses rilis ini memungkinkan Anda dengan cepat dan andal membuat perubahan pada AWS infrastruktur Anda.

Misalnya, Anda dapat membuat alur kerja yang secara otomatis membangun tumpukan uji saat Anda mengirimkan templat yang diperbarui ke repositori kode. Setelah CloudFormation membangun tumpukan pengujian, Anda dapat mengujinya dan kemudian memutuskan apakah akan mendorong perubahan ke tumpukan produksi. Untuk informasi lebih lanjut tentang manfaat pengiriman berkelanjutan, [Apa yang dimaksud dengan pengiriman berkelanjutan?](#).

Gunakan CodePipeline untuk membangun alur kerja pengiriman berkelanjutan dengan membangun pipa untuk CloudFormation tumpukan. CodePipeline memiliki integrasi bawaan dengan CloudFormation, sehingga Anda dapat menentukan tindakan CloudFormation -spesifik, seperti membuat, memperbarui, atau menghapus tumpukan, dalam pipeline. Untuk informasi selengkapnya CodePipeline, lihat [Panduan AWS CodePipeline Pengguna](#).

Topik

- [Panduan: Membangun alur untuk tumpukan uji dan produksi](#)
- [AWS CloudFormation referensi properti konfigurasi](#)
- [AWS CloudFormation artefak](#)
- [Menggunakan fungsi penggantian parameter dengan saluran pipa CodePipeline](#)

Panduan: Membangun alur untuk tumpukan uji dan produksi

Bayangkan proses rilis di mana Anda mengirimkan AWS CloudFormation template, yang CloudFormation kemudian digunakan untuk secara otomatis membangun tumpukan pengujian. Setelah Anda meninjau tumpukan uji, Anda dapat melihat pratinjau bagaimana perubahan akan memodifikasi tumpukan produksi Anda, dan kemudian memilih apakah akan menerapkannya. Untuk menyelesaikan alur kerja ini, Anda dapat menggunakan CloudFormation untuk membangun tumpukan pengujian, menghapus tumpukan pengujian, membuat set perubahan, dan kemudian menjalankan set perubahan. Namun, dengan setiap tindakan, Anda perlu berinteraksi secara manual CloudFormation. Dalam panduan ini, kami akan membangun CodePipeline pipeline

yang mengotomatiskan banyak tindakan ini, membantu Anda mencapai alur kerja pengiriman berkelanjutan dengan tumpukan Anda. CloudFormation

Prasyarat

Panduan ini mengasumsikan bahwa Anda telah menggunakan CodePipeline dan CloudFormation, dan mengetahui cara kerja pipa dan AWS CloudFormation templat serta tumpukan. Untuk informasi selengkapnya CodePipeline, lihat [Panduan AWS CodePipeline Pengguna](#). Anda juga harus memiliki bucket Amazon S3 yang sama seperti saat Wilayah AWS Anda membuat pipeline.

Important

WordPress Template sampel membuat EC2 instance yang memerlukan koneksi ke Internet. Periksa apakah Anda memiliki VPC default dan subnet yang mengizinkan lalu lintas ke Internet.

Gambaran umum panduan

Panduan ini membangun pipeline untuk WordPress situs sampel dalam tumpukan. Alur dipisahkan menjadi tiga tahap. Setiap tahap harus berisi setidaknya satu tindakan, yang merupakan tugas alur pada artifact Anda (masukan Anda). Tahap mengatur tindakan dalam alur. CodePipeline harus menyelesaikan semua tindakan dalam tahap sebelum tahap tersebut memproses artifact baru, misalnya, jika Anda mengirimkan masukan baru untuk menjalankan kembali alur.

Pada akhir panduan ini, Anda akan memiliki alur yang melakukan alur kerja berikut:

1. Tahap pertama dari pipeline mengambil artefak sumber (CloudFormation template dan file konfigurasinya) dari repositori.

Anda akan menyiapkan artefak yang menyertakan WordPress template sampel dan mengunggahnya ke bucket S3.

2. Pada tahap kedua, alur membuat tumpukan uji dan kemudian menunggu persetujuan Anda.

Setelah Anda meninjau tumpukan uji, Anda dapat memilih untuk melanjutkan dengan alur asli atau membuat dan mengirimkan artifact lain untuk membuat perubahan. Jika Anda menyetujui, tahap ini menghapus tumpukan uji, dan kemudian alur berlanjut ke tahap berikutnya.

3. Pada tahap ketiga, alur menciptakan perubahan yang ditetapkan terhadap tumpukan produksi, dan kemudian menunggu persetujuan Anda.

Di awal, Anda tidak akan memiliki tumpukan produksi. Set perubahan menunjukkan kepada Anda semua sumber daya yang CloudFormation akan dibuat. Jika Anda menyetujui, tahap ini mengeksekusi perubahan set dan membangun tumpukan produksi Anda.

Note

CloudFormation adalah layanan gratis. Namun, Anda dikenakan biaya untuk AWS sumber daya, seperti EC2 instance, yang Anda sertakan dalam tumpukan Anda dengan tarif saat ini untuk masing-masing sumber daya. Untuk informasi lebih lanjut tentang AWS harga, lihat halaman detail untuk setiap produk di <http://aws.amazon.com>.

Langkah 1: Edit artifact dan unggah ke Bucket S3

Sebelum membangun pipeline, Anda harus menyiapkan repositori sumber dan file.

CodePipelinenyalin file sumber ini ke [penyimpanan artefak](#) pipeline Anda, lalu menggunakannya untuk melakukan tindakan di pipeline Anda, seperti membuat CloudFormation tumpukan.

Saat Anda menggunakan Amazon Simple Storage Service (Amazon S3) sebagai repositori sumber CodePipeline, Anda harus mem-zip file sumber sebelum mengunggahnya ke bucket S3. File zip adalah CodePipeline artefak yang dapat berisi CloudFormation template, file konfigurasi template, atau keduanya. Kami menyediakan artefak yang berisi WordPress template sampel dan dua file konfigurasi template. Kedua file konfigurasi menentukan nilai parameter untuk WordPress template. CodePipeline menggunakannya saat membuat WordPress tumpukan. Satu file berisi nilai parameter untuk tumpukan tes, dan yang lainnya untuk tumpukan produksi. Anda harus mengedit file konfigurasi, misalnya, untuk menentukan nama EC2 pasangan kunci yang ada yang Anda miliki. Untuk informasi lebih lanjut tentang artifact, lihat [AWS CloudFormation artefak](#).

Setelah Anda membuat artifact, Anda akan mengunggahnya ke bucket S3.

Untuk mengedit dan mengunggah artifact

1. Unduh dan buka artefak sampel: <https://s3.amazonaws.com/cloudformation-examples/user-guide/continuous-deployment/wordpress-single-instance.zip>.

Artefak berisi tiga file:

- WordPress Template sampel: `wordpress-single-instance.yaml`

- File konfigurasi templat untuk tumpukan uji.: `test-stack-configuration.json`
 - File konfigurasi templat untuk tumpukan produksi: `prod-stack-configuration.json`
2. Ekstrak semua file, dan kemudian gunakan editor teks untuk mengubah file konfigurasi templat.

Buka file konfigurasi untuk melihat bahwa file tersebut berisi pasangan kunci-nilai yang dipetakan ke parameter WordPress template. File konfigurasi menentukan nilai parameter yang menggunakan alur saat membuat tumpukan uji dan produksi.

Edit `test-stack-configuration.json` file untuk menentukan nilai parameter untuk tumpukan uji dan `prod-stack-configuration.json` file untuk tumpukan produksi.

- Ubah nilai `DBPassword` dan `DBRootPassword` kunci ke kata sandi yang dapat Anda gunakan untuk masuk ke WordPress database Anda. Seperti yang didefinisikan dalam WordPress template, nilai parameter harus berisi hanya karakter alfanumerik.
 - Ubah nilai `KeyName` kunci ke nama EC2 pasangan kunci yang ada di wilayah tempat Anda akan membuat pipeline.
3. Tambahkan file konfigurasi yang diubah ke artifact file (`.zip`) asli, mengganti file duplikat.

Anda sekarang memiliki artifact disesuaikan dan yang dapat Anda unggah ke Bucket S3.

4. [Unggah artefak ke bucket S3 yang Anda miliki.](#)

Perhatikan lokasi file. Anda akan menentukan lokasi file ini saat membangun alur.

Catatan tentang artifact dan Bucket S3:

- Gunakan bucket yang berada di AWS Wilayah yang sama di mana Anda akan membuat pipeline Anda.
- CodePipeline mengharuskan bucket mengaktifkan [versi](#).
- Anda juga dapat menggunakan layanan yang tidak mengharuskan Anda untuk zip file Anda sebelum mengunggahnya, seperti GitHub atau CodeCommit, untuk repositori sumber Anda.
- Artifact dapat berisi informasi sensitif seperti kata sandi. Batas akses sehingga hanya pengguna yang diizinkan yang dapat melihat file. Ketika Anda melakukannya, pastikan bahwa masih CodePipeline dapat mengakses file. Misalnya, jika Anda mengunggah artifact ke bucket S3, gunakan [Kebijakan atau kebijakan pengguna bucket S3](#) untuk membatasi akses.

Anda sekarang memiliki artefak yang CodePipeline dapat menarik ke pipa Anda. Pada langkah berikutnya, Anda akan menentukan lokasi artefak dan membangun WordPress pipa.

Langkah 2: Buat tumpukan alur

Untuk membuat WordPress pipeline, Anda akan menggunakan CloudFormation template sampel. Selain membangun pipeline, template menyiapkan peran layanan AWS Identity and Access Management (IAM) untuk CodePipeline dan CloudFormation, bucket S3 untuk penyimpanan CodePipeline artefak, dan topik Amazon Simple Notification Service (Amazon SNS) tempat pipeline mengirimkan notifikasi, seperti notifikasi tentang ulasan. Template sampel memudahkan penyediaan dan konfigurasi sumber daya ini dalam satu CloudFormation tumpukan.

Untuk detail lebih lanjut tentang konfigurasi alur, lihat [Apa yang dilakukan alur](#).

Important

WordPress Template sampel membuat EC2 instance yang memerlukan koneksi ke Internet. Periksa apakah VPC dan subnet default Anda mengizinkan lalu lintas ke Internet.

Cara membuat tumpukan alur

1. Unduh template sampel di <https://s3.amazonaws.com/cloudformation-examples/user-guide/continuous-deployment/basic-pipeline.yml>. Simpan di komputer Anda.
2. Buka CloudFormation konsol di <https://console.aws.amazon.com/cloudformation/>.
3. Pilih AWS Wilayah yang mendukung CodePipeline dan CloudFormation.

Untuk informasi selengkapnya, lihat [CodePipeline titik akhir dan kuota serta AWS CloudFormation titik akhir dan kuota](#) di. Referensi Umum AWS

4. Pilih Buat tumpukan.
5. Di bagian Tentukan templat, pilih Unggah file templat, lalu pilih templat yang baru saja diunduh, `basic-pipeline.yml`.
6. Pilih Berikutnya.
7. Untuk Nama tumpukan, ketik `sample-WordPress-pipeline`.
8. Di bagian Parameter, tentukan nilai parameter berikut, dan kemudian pilih Selanjutnya. Saat mengatur parameter tumpukan, jika Anda menyimpan nama yang sama untuk WordPress template dan file konfigurasinya, Anda dapat menggunakan nilai default. Jika tidak, tentukan nama file yang Anda gunakan.

PipelineName

Nama alur Anda, seperti `WordPress-test-pipeline`.

S3Bucket

Nama bucket S3 tempat Anda menyimpan artefak (.zip).

Sources3key

Nama file artefak Anda. Jika Anda menyimpan artefak dalam folder, sertakan sebagai bagian dari nama file, seperti `folder/subfolder/wordpress-single-instance.zip`.

Email

Alamat email yang CodePipeline mengirimkan pemberitahuan pipeline, seperti `myemail@example.com`.

9. Untuk panduan ini, Anda tidak perlu menambahkan tanda atau menentukan pengaturan lanjutan, jadi pilih Selanjutnya.
10. Pastikan bahwa nama tumpukan dan templat URL sudah benar, dan kemudian pilih Buat tumpukan.
11. Untuk mengetahui bahwa Anda mengetahui bahwa CloudFormation mungkin membuat sumber daya IAM, pilih kotak centang.

Mungkin perlu beberapa menit AWS CloudFormation untuk membuat tumpukan Anda. Untuk memantau kemajuan, tampilkan peristiwa tumpukan. Untuk informasi selengkapnya, lihat [Pantau kemajuan tumpukan](#).

Setelah tumpukan Anda dibuat, CodePipeline mulai pipeline baru Anda. Untuk melihat statusnya, lihat [CodePipeline konsol](#). Dari daftar pipa, pilih `WordPress-test-pipeline`.

Apa yang dilakukan alur

Bagian ini menjelaskan tiga tahap pipeline, menggunakan cuplikan dari template WordPress pipeline sampel.

Tahap 1: Sumber

Tahap pertama alur adalah tahap sumber di mana Anda menentukan lokasi kode sumber Anda. Setiap kali Anda mendorong revisi ke lokasi ini, CodePipeline jalankan ulang pipa Anda.

Kode sumber terletak di bucket S3 dan diidentifikasi dengan nama filenya. Anda menetapkan nilai-nilai ini sebagai nilai parameter masukan ketika Anda membuat tumpukan alur. Untuk mengizinkan penggunaan Artifact sumber dalam tahap berikutnya, potongan menentukan `OutputArtifacts` properti, dengan nama `TemplateSource`. Untuk menggunakan Artifact ini pada tahap selanjutnya, Anda tentukan `TemplateSource` sebagai Artifact masukan.

```
- Name: S3Source
  Actions:
    - Name: TemplateSource
      ActionTypeId:
        Category: Source
        Owner: AWS
        Provider: S3
        Version: '1'
      Configuration:
        S3Bucket: !Ref 'S3Bucket'
        S3ObjectKey: !Ref 'SourceS3Key'
      OutputArtifacts:
        - Name: TemplateSource
```

Tahap 2: TestStage

Di `TestStage` tahap, alur menciptakan tumpukan uji, menunggu persetujuan, dan kemudian menghapus tumpukan uji.

Untuk `CreateStack` tindakan, pipeline menggunakan file konfigurasi pengujian dan WordPress template untuk membuat tumpukan pengujian. Kedua file yang terkandung dalam `TemplateSource` Artifact input, yang dibawa dari tahap sumber. Potongan menggunakan `REPLACE_ON_FAILURE` mode tindakan. Jika pembuatan tumpukan gagal, alur menggantikannya sehingga Anda tidak perlu membersihkan atau memecahkan masalah tumpukan sebelum Anda dapat menjalankan alur kembali. Modus aksi berguna untuk iterasi cepat pada tumpukan uji. Untuk `RoleArn` properti, nilainya adalah peran AWS CloudFormation layanan yang dideklarasikan di tempat lain dalam template.

Tindakan `ApproveTestStack` menjeda alur dan mengirimkan notifikasi ke alamat email yang telah Anda tentukan saat membuat tumpukan pipa. Saat pipeline dijeda, Anda dapat memeriksa tumpukan WordPress pengujian dan sumber dayanya. Gunakan CodePipeline untuk [menyetujui atau menolak tindakan](#) ini. Properti `CustomData` berisi deskripsi tindakan yang Anda setujui, yang ditambahkan alur ke email pemberitahuan.

Setelah Anda menyetujui tindakan ini, CodePipeline pindah ke DeleteTestStack tindakan dan menghapus WordPress tumpukan pengujian dan sumber dayanya.

```
- 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'
```

Tahap 3: ProdStage

Tahap ProdStage alur menciptakan perubahan yang ditetapkan terhadap tumpukan produksi yang ada, menunggu persetujuan, dan kemudian mengeksekusi set perubahan.

Set perubahan menyediakan pratinjau semua modifikasi yang AWS CloudFormation akan dilakukan pada tumpukan produksi Anda sebelum menerapkannya. Saat menjalankan alur pertama Anda, Anda tidak akan memiliki tumpukan produksi yang berjalan. Set perubahan menunjukkan tindakan yang AWS CloudFormation dilakukan saat membuat tumpukan pengujian. Untuk membuat set perubahan, CreateChangeSet tindakan menggunakan template WordPress sampel dan konfigurasi template produksi dari artefak TemplateSource input.

Mirip dengan tahap sebelumnya, tindakanApproveChangeSet menjeda alur dan mengirimkan pemberitahuan email. Saat pipeline dijeda, Anda dapat melihat set perubahan untuk memeriksa semua modifikasi yang diusulkan pada tumpukan produksi WordPress . Gunakan CodePipeline untuk [menyetujui atau menolak](#) tindakan ini untuk melanjutkan atau menghentikan pipeline, masing-masing.

Setelah Anda menyetujui tindakan ini, ExecuteChangeSet tindakan akan mengeksekusi set perubahan, sehingga AWS CloudFormation melakukan semua tindakan yang dijelaskan dalam set perubahan. Untuk proses awal, AWS CloudFormation buat tumpukan WordPress produksi. Saat menjalankan berikutnya, AWS CloudFormation memperbarui tumpukan.

```
- 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'
```

Langkah 3: Lihat WordPress tumpukan

Saat CodePipeline berjalan melalui pipa, ia digunakan CloudFormation untuk membuat tumpukan pengujian dan produksi. Untuk melihat status tumpukan ini dan outputnya, gunakan CloudFormation konsol.

Untuk melihat tumpukan

1. Buka CloudFormation konsol di <https://console.aws.amazon.com/cloudformation/>.
2. Tergantung pada apakah alur sedang dalam tahap pengujian atau produksi, pilih Test-MyWordPressSite atau Prod-MyWordPressSite tumpukan.
3. Untuk memeriksa status tumpukan Anda, lihat tumpukan [peristiwa](#).

Jika tumpukan dalam keadaan gagal, lihat alasan status untuk menemukan kesalahan tumpukan. Perbaiki kesalahan, dan kemudian jalankan kembali alur. Jika tumpukan dalam CREATE_COMPLETE keadaan, lihat outputnya untuk mendapatkan URL WordPress situs Anda.

Anda telah berhasil digunakan CodePipeline untuk membangun alur kerja pengiriman berkelanjutan untuk WordPress situs sampel. Jika Anda mengirimkan perubahan ke bucket S3, CodePipeline secara otomatis mendeteksi versi baru, lalu menjalankan ulang pipeline Anda. Alur kerja ini memudahkan untuk mengirimkan dan menguji perubahan sebelum membuat perubahan pada situs produksi Anda.

Langkah 4: Bersihkan Sumber Daya

Untuk memastikan bahwa Anda tidak dikenakan biaya untuk layanan yang tidak diinginkan, hapus sumber daya Anda.

Important

Hapus WordPress tumpukan pengujian dan produksi sebelum menghapus tumpukan pipa. Tumpukan pipeline berisi peran layanan yang diperlukan untuk menghapus WordPress tumpukan. Jika Anda menghapus tumpukan pipeline terlebih dahulu, Anda dapat mengaitkan peran layanan lain Amazon Resource Name (ARN) dengan WordPress tumpukan, lalu menghapusnya.

Untuk menghapus objek di penyimpanan Artifact

1. Buka konsol Amazon S3 di <https://console.aws.amazon.com/s3/>
2. Pilih bucket S3 yang CodePipeline digunakan sebagai toko artefak pipa Anda.

Nama bucket mengikuti format: *stackname-artifactstorebucket-id*. Jika Anda mengikuti panduan ini, nama bucket mungkin terlihat mirip dengan contoh berikut: `sample-WordPress-pipeline-artifactstorebucket-12345abcd12345`.

3. Hapus semua objek di penyimpanan Artifact Bucket S3.

Bila Anda menghapus tumpukan alur pada langkah berikutnya, bucket ini harus kosong. Jika CloudFormation tidak, tidak akan dapat menghapus ember.

Untuk menghapus tumpukan Anda

1. Dari CloudFormation konsol, pilih tumpukan yang ingin Anda hapus.

Jika WordPress tumpukan yang dibuat oleh pipa masih berjalan, pilih terlebih dahulu. Secara default, nama tumpukan `Test-MyWordPressSite` dan `Prod-MyWordPressSite`.

Jika Anda sudah menghapus WordPress tumpukan, pilih `sample-WordPress-pipeline` tumpukan.

2. Pilih Tindakan, dan kemudian pilih Hapus Tumpukan.
3. Pada pesan konfirmasi, pilih Ya, Hapus.

CloudFormation menghapus tumpukan semua sumber daya tumpukan, seperti EC2 instance, topik notifikasi, peran layanan, dan pipeline.

Sekarang setelah Anda memahami cara membangun CloudFormation alur kerja dasar CodePipeline, Anda dapat menggunakan contoh template dan artefak sebagai titik awal untuk membangun sendiri.

Lihat juga

Sumber daya terkait berikut dapat membantu Anda saat menggunakan parameter ini.

- Untuk informasi selengkapnya tentang parameter CloudFormation tindakan CodePipeline, lihat [referensi konfigurasi tindakan AWS CloudFormation penerapan](#) di Panduan AWS CodePipeline Pengguna.
- Misalnya nilai templat menurut penyedia tindakan, seperti untuk Owner bidang atau configuration bidang, lihat [referensi struktur tindakan](#) di Panduan AWS CodePipeline Pengguna.
- Untuk mengunduh contoh templat tumpukan pipeline dalam format YAMAL atau JSON, lihat [Tutorial: Membuat pipeline dengan AWS CloudFormation](#) [AWS CodePipeline](#) Panduan Pengguna.

AWS CloudFormation referensi properti konfigurasi

Saat Anda membangun CodePipeline pipeline, Anda menambahkan Deploy tindakan ke pipeline dengan AWS CloudFormation sebagai penyedia. Anda kemudian harus menentukan AWS CloudFormation tindakan mana yang dipanggil pipeline dan pengaturan tindakan. Topik ini menjelaskan properti konfigurasi AWS CloudFormation . Untuk menentukan properti, Anda dapat menggunakan CodePipeline konsol, atau Anda dapat membuat objek JSON yang akan digunakan untuk AWS CLI, CodePipeline API, atau AWS CloudFormation templat.

Topik

- [Properti konfigurasi \(konsol\)](#)

- [Properti konfigurasi \(objek JSON\)](#)
- [Lihat juga](#)

Properti konfigurasi (konsol)

CodePipeline [Konsol](#) menampilkan properti konfigurasi dan menunjukkan properti yang diperlukan berdasarkan mode tindakan yang Anda pilih.

Note

Bila Anda membuat alur, Anda dapat menentukan mode aksi Membuat atau memperbarui tumpukan atau Membuat atau mengganti satu set perubahan saja. Properti di bagian Lanjutan tersedia hanya ketika Anda mengedit alur.

Mode aksi

AWS CloudFormation Tindakan yang CodePipeline dipanggil saat memproses tahap terkait. Pilih salah satu mode tindakan berikut ini:

- Membuat atau mengganti satu set perubahan membuat set perubahan jika tidak ada berdasarkan nama tumpukan dan template yang Anda kirimkan. Jika set perubahan ada, AWS CloudFormation menghapusnya, dan kemudian membuat yang baru.
- Membuat atau memperbarui tumpukan membuat tumpukan jika tumpukan yang ditentukan tidak ada. Jika tumpukan ada, AWS CloudFormation perbarui tumpukan. Gunakan tindakan ini untuk memperbarui tumpukan yang ada. CodePipeline tidak akan menggantikan tumpukan.
- Menghapus tumpukan akan menghapus tumpukan. Jika Anda menentukan tumpukan yang tidak ada, tindakan berhasil diselesaikan tanpa menghapus tumpukan.
- Mengeksekusi set perubahan akan mengeksekusi satu set perubahan.
- Ganti tumpukan gagal membuat tumpukan jika tumpukan yang ditentukan tidak ada. Jika tumpukan ada dan dalam keadaan gagal (dilaporkan sebagai `ROLLBACK_COMPLETE`, `ROLLBACK_FAILED`, `CREATE_FAILED`, `DELETE_FAILED`, atau `UPDATE_ROLLBACK_FAILED`), AWS CloudFormation menghapus tumpukan dan kemudian membuat yang baru. Jika tumpukan tidak dalam keadaan gagal, AWS CloudFormation perbarui. Gunakan tindakan ini untuk mengganti tumpukan yang gagal tanpa memulihkan atau memecahkan masalahnya. Anda biasanya akan memilih mode ini untuk pengujian.

Nama tumpukan

Nama yang terkait dengan tumpukan yang ada atau tumpukan yang ingin Anda buat. Nama harus unik di AWS Wilayah tempat Anda membuat tumpukan.

Note

Nama tumpukan hanya dapat berisi karakter alfanumerik (peka huruf besar dan kecil) dan tanda hubung. Nama itu harus dimulai dengan karakter abjad dan tidak boleh lebih dari 128 karakter.

Ubah nama set

Nama set perubahan yang ada atau set perubahan baru yang ingin Anda buat untuk tumpukan tertentu.

Template

Lokasi file AWS CloudFormation template, yang mengikuti format *ArtifactName::TemplateFileName*.

Konfigurasi template

Lokasi file konfigurasi template, yang mengikuti format *ArtifactName::TemplateConfigurationFileName*. File konfigurasi templat dapat berisi nilai parameter templat, kebijakan tumpukan, dan tag. Jika Anda menyertakan informasi sensitif, seperti kata sandi, batasi akses ke file ini. Untuk informasi selengkapnya, lihat [AWS CloudFormation artefak](#).

Kemampuan

Untuk tumpukan yang berisi sumber daya tertentu, pengakuan eksplisit yang AWS CloudFormation mungkin membuat atau memperbarui sumber daya tersebut. Misalnya, Anda harus menentukan `CAPABILITY_IAM` apakah template tumpukan Anda berisi sumber daya AWS Identity and Access Management (IAM). Untuk informasi selengkapnya, silakan lihat [CreateStack](#) Parameter permintaan operasi API.

Jika Anda memiliki sumber daya IAM di template tumpukan Anda, Anda harus menentukan properti ini.

Anda dapat menentukan lebih dari satu kemampuan.

Nama peran

Nama peran layanan IAM yang AWS CloudFormation mengasumsikan ketika beroperasi pada sumber daya dalam tumpukan yang ditentukan.

Nama berkas keluaran

Di bagian Lanjutan, Anda dapat menentukan nama file keluaran, seperti `CreateStackOutput.json`, yang CodePipeline menambah [artefak keluaran](#) setelah melakukan tindakan yang ditentukan. Artefak keluaran berisi file JSON dengan isi `Outputs` bagian template. AWS CloudFormation

Jika Anda tidak menentukan nama, CodePipeline tidak menghasilkan artefak keluaran.

Penggantian parameter

Parameter ditetapkan di templat Anda dan memungkinkan Anda untuk memasukkan nilai kustom saat membuat atau memperbarui tumpukan. Anda dapat menentukan sebuah objek JSON yang menimpa nilai parameter templat dalam file konfigurasi templat. Semua nama parameter harus ada dalam templat tumpukan. Untuk informasi selengkapnya, lihat [CloudFormation Parameters sintaks template](#).

Note

Ada batas ukuran maksimum 1 kilobyte untuk objek JSON yang dapat disimpan dalam properti `ParameterOverrides`.

Kami merekomendasikan agar Anda menggunakan file konfigurasi templat untuk menentukan sebagian besar nilai parameter Anda. Gunakan penimpaan parameter untuk menentukan hanya nilai parameter dinamis. Parameter dinamis tidak diketahui sebelum Anda menjalankan alur.

Contoh berikut menetapkan nilai untuk parameter `ParameterName` menggunakan fungsi penimpaan parameter. Fungsi mengambil nilai dari artefak CodePipeline input. Untuk informasi lebih lanjut tentang fungsi penimpaan parameter, lihat [Menggunakan fungsi penggantian parameter dengan saluran pipa CodePipeline](#).

```
{
  "ParameterName" : { "Fn::GetParam" : ["ArtifactName", "config-file-name.json",
    "ParamName"]}
}
```

```
}
```

Properti konfigurasi (objek JSON)

Bila Anda menentukan CloudFormation sebagai penyedia untuk tindakan tahap, menentukan properti berikut di properti Configuration. Gunakan objek JSON untuk AWS CLI, CodePipeline API, atau AWS CloudFormation template. Sebagai contoh, lihat [Panduan: Membangun alur untuk tumpukan uji dan produksi](#) dan [AWS CloudFormation referensi properti konfigurasi](#).

ActionMode

AWS CloudFormation Tindakan yang CodePipeline dipanggil ketika memproses tahap terkait. Tentukan hanya salah satu mode tindakan berikut:

- **CHANGE_SET_EXECUTE** mengeksekusi set perubahan.
- **CHANGE_SET_REPLACE** membuat set perubahan, jika tidak ada, berdasarkan nama tumpukan dan templat yang Anda kirimkan. Jika set perubahan ada, AWS CloudFormation menghapusnya, dan kemudian membuat yang baru.
- **CREATE_UPDATE** membuat tumpukan jika tumpukan yang ditentukan tidak ada. Jika tumpukan ada, AWS CloudFormation akan memperbarui tumpukan. Gunakan tindakan ini untuk memperbarui tumpukan yang ada. CodePipeline tidak akan menggantikan tumpukan.
- **DELETE_ONLY** akan menghapus tumpukan. Jika Anda menentukan tumpukan yang tidak ada, tindakan berhasil diselesaikan tanpa menghapus tumpukan.
- **REPLACE_ON_FAILURE** membuat tumpukan, jika tumpukan yang ditentukan tidak ada. Jika tumpukan ada dan dalam keadaan gagal (dilaporkan sebagai **ROLLBACK_COMPLETE**, **ROLLBACK_FAILED**, **CREATE_FAILED**, **DELETE_FAILED**, atau **UPDATE_ROLLBACK_FAILED**), AWS CloudFormation menghapus tumpukan dan kemudian membuat tumpukan baru. Jika tumpukan tidak dalam keadaan gagal, AWS CloudFormation perbarui. Gunakan tindakan ini untuk secara otomatis mengganti tumpukan yang gagal tanpa memulihkan atau memecahkan masalahnya. Anda biasanya akan memilih mode ini untuk pengujian.

Properti ini diperlukan.

Capabilities

Untuk tumpukan yang berisi sumber daya tertentu, pengakuan eksplisit yang AWS CloudFormation mungkin membuat atau memperbarui sumber daya tersebut. Misalnya, Anda

harus menentukan `CAPABILITY_IAM` apakah template tumpukan Anda berisi sumber daya AWS Identity and Access Management (IAM). Untuk informasi selengkapnya, silakan lihat [CreateStack](#) Parameter permintaan operasi API.

Properti ini bersyarat. Jika Anda memiliki sumber daya IAM di template tumpukan Anda, Anda harus menentukan properti ini.

Anda dapat menentukan beberapa kemampuan. Contoh berikut menambahkan `CAPABILITY_IAM` dan `CAPABILITY_AUTO_EXPAND` properti ke template:

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

Nama set perubahan yang ada atau set perubahan baru yang ingin Anda buat untuk tumpukan tertentu.

Properti ini diperlukan untuk mode tindakan berikut: `CHANGE_SET_REPLACE` dan `CHANGE_SET_EXECUTE`. Untuk semua mode tindakan lainnya, properti ini diabaikan.

OutputFileName

Sebuah nama untuk file output, seperti `CreateStackOutput.json`. CodePipeline menambahkan file ke [artefak output](#) setelah melakukan tindakan yang ditentukan. Artefak keluaran berisi file JSON dengan isi `Outputs` bagian template. AWS CloudFormation

Properti ini adalah opsional. Jika Anda tidak menentukan nama, CodePipeline tidak menghasilkan artefak keluaran.

ParameterOverrides

Parameter ditetapkan di templat Anda dan memungkinkan Anda untuk memasukkan nilai kustom saat membuat atau memperbarui tumpukan. Anda dapat menentukan sebuah objek JSON yang menimpa nilai parameter templat dalam file konfigurasi templat. Semua nama parameter harus ada dalam templat tumpukan. Untuk informasi selengkapnya, lihat [CloudFormation Parameters sintaks template](#).

Contoh berikut menambahkan penyimpanan parameter `InstanceType` dan `KeyName` ke templat: **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

Ukuran maksimum untuk objek JSON yang dapat disimpan dalam properti `ParameterOverrides` adalah 1 kilobyte.

Kami merekomendasikan agar Anda menggunakan file konfigurasi templat untuk menentukan sebagian besar nilai parameter Anda. Gunakan penyimpanan parameter untuk menentukan hanya nilai parameter dinamis. Nilai parameter dinamis tidak diketahui sampai Anda menjalankan alur.

Contoh berikut menetapkan nilai untuk parameter `ParameterName` menggunakan fungsi penyimpanan parameter. Fungsi mengambil nilai dari artefak `CodePipeline` input. Untuk informasi lebih lanjut tentang fungsi penyimpanan parameter, lihat [Menggunakan fungsi penggantian parameter dengan saluran pipa CodePipeline](#).

```
{
  "ParameterName" : { "Fn::GetParam" : ["ArtifactName", "config-file-name.json",
    "ParamName"]}
}
```

Properti ini adalah opsional.

RoleArn

Nama Sumber Daya Amazon (ARN) dari peran layanan IAM yang AWS CloudFormation mengasumsikan ketika beroperasi pada sumber daya dalam tumpukan.

Properti ini diperlukan untuk mode tindakan berikut: `CREATE_UPDATE`, `REPLACE_ON_FAILURE`, `DELETE_ONLY`, dan `CHANGE_SET_REPLACE`. `RoleArn` tidak diterapkan ketika mengeksekusi set perubahan. Jika Anda tidak menggunakan `CodePipeline` untuk membuat set perubahan, pastikan bahwa set perubahan atau tumpukan memiliki peran terkait.

StackName

Nama tumpukan yang ada atau tumpukan yang ingin Anda buat.

Properti ini diperlukan untuk semua mode tindakan.

TemplateConfiguration

TemplateConfiguration adalah file konfigurasi templat. Anda menyertakan file dalam artefak input untuk tindakan ini. Nama file konfigurasi template mengikuti format ini:

Artifactname::TemplateConfigurationFileName

Artifactname adalah nama artefak input seperti yang muncul di CodePipeline. Sebagai contoh, tahap sumber dengan nama artefak SourceArtifact dan nama file test-configuration.json membuat nama TemplateConfiguration seperti yang ditunjukkan dalam contoh ini:

```
"TemplateConfiguration": "SourceArtifact::test-configuration.json"
```

File konfigurasi templat dapat berisi nilai parameter template dan kebijakan tumpukan. Jika Anda menyertakan informasi sensitif, seperti kata sandi, batasi akses ke file ini. Untuk contoh file konfigurasi template, lihat [AWS CloudFormation artefak](#).

Properti ini adalah opsional.

TemplatePath

TemplatePath mewakili file AWS CloudFormation template. Anda menyertakan file dalam artefak input untuk tindakan ini. Nama file mengikuti format ini:

Artifactname::TemplateFileName

Artifactname adalah nama artefak input seperti yang muncul di CodePipeline. Sebagai contoh, tahap sumber dengan nama artefak SourceArtifact dan nama file template.yaml membuat nama TemplatePath seperti yang ditunjukkan dalam contoh ini:

```
"TemplatePath": "SourceArtifact::template.yaml"
```

Properti ini diperlukan untuk mode tindakan berikut: CREATE_UPDATE, REPLACE_ON_FAILURE, dan CHANGE_SET_REPLACE. Untuk semua mode tindakan lainnya, properti ini diabaikan.

Lihat juga

Sumber daya terkait berikut dapat membantu Anda saat menggunakan parameter ini.

- Untuk informasi selengkapnya tentang parameter CloudFormation tindakan CodePipeline, lihat [referensi konfigurasi tindakan AWS CloudFormation penerapan](#) di Panduan AWS CodePipeline Pengguna.
- Misalnya nilai templat menurut penyedia tindakan, seperti untuk Owner bidang atau configuration bidang, lihat [referensi struktur tindakan](#) di Panduan AWS CodePipeline Pengguna.
- Untuk mengunduh contoh templat tumpukan pipeline dalam format YAMAL atau JSON, lihat [Tutorial: Membuat pipeline dengan AWS CloudFormation](#) di Panduan Pengguna.

AWS CloudFormation artefak

CodePipeline melakukan tugas-tugas pada artefak seperti CodePipeline menjalankan pipa. Untuk AWS CloudFormation, artefak dapat menyertakan file template tumpukan, file konfigurasi template, atau keduanya. CodePipeline menggunakan artefak ini untuk bekerja dengan AWS CloudFormation tumpukan dan mengubah set.

Jika Anda menggunakan Amazon Simple Storage Service (Amazon S3) sebagai repositori sumber, Anda harus zip file konfigurasi template dan template ke dalam satu file sebelum Anda mengunggahnya ke bucket S3. Untuk repositori lain, seperti GitHub dan AWS CodeCommit, unggah artefak tanpa ritsleting. Untuk informasi selengkapnya, lihat [Membuat pipeline, tahapan, dan tindakan](#) di Panduan AWS CodePipeline Pengguna.

Anda dapat menambahkan sebanyak mungkin file yang Anda butuhkan ke repositori Anda. Misalnya, Anda mungkin ingin menyertakan dua konfigurasi yang berbeda untuk templat yang sama: satu untuk konfigurasi pengujian dan satu lagi untuk konfigurasi produksi.

Topik ini menjelaskan setiap jenis artefact.

Topik

- [File templat tumpukan](#)
- [File konfigurasi templat](#)
- [Lihat juga](#)

File templat tumpukan

File template tumpukan mendefinisikan sumber daya yang menyediakan dan AWS CloudFormation mengonfigurasi. File-file ini adalah file template yang sama yang Anda gunakan saat Anda membuat

atau memperbarui tumpukan menggunakan AWS CloudFormation. Anda dapat menggunakan tempat berformat YAML atau JSON. Untuk informasi lebih lanjut tentang templat, lihat [CloudFormation bagian template](#).

File konfigurasi templat

File konfigurasi templat adalah file teks berformat JSON yang dapat menentukan nilai parameter templat, [kebijakan tumpukan](#), dan tanda. Gunakan file konfigurasi ini untuk menentukan nilai parameter atau kebijakan tumpukan untuk tumpukan. Semua nilai parameter yang Anda tentukan harus dinyatakan di templat terkait.

Jika Anda menyertakan informasi sensitif—seperti kata sandi—dalam file ini, batasi akses ke file tersebut. Misalnya, jika Anda mengunggah artifact ke bucket S3, gunakan [Kebijakan atau kebijakan pengguna bucket S3](#) untuk membatasi akses.

Untuk membuat file konfigurasi, gunakan format berikut:

```
{
  "Parameters" : {
    "NameOfTemplateParameter" : "ValueOfParameter",
    ...
  },
  "Tags" : {
    "TagKey" : "TagValue",
    ...
  },
  "StackPolicy" : {
    "Statement" : [
      StackPolicyStatement
    ]
  }
}
```

Contoh berikut menentukan TestEC2Key untuk parameter KeyName, menambahkan tanda Department yang nilainya Marketing, dan menambahkan kebijakan tumpukan yang memungkinkan semua tindakan pembaruan kecuali pembaruan yang menghapus sumber daya.

```
{
  "Parameters" : {
    "KeyName" : "TestEC2Key"
  },
```

```
"Tags" : {
  "Department" : "Marketing"
},
"StackPolicy" : {
  "Statement" : [
    {
      "Effect" : "Allow",
      "NotAction" : "Update:Delete",
      "Principal": "*",
      "Resource" : "*"
    }
  ]
}
```

Lihat juga

Sumber daya terkait berikut dapat membantu Anda saat menggunakan parameter ini.

- Untuk informasi selengkapnya tentang parameter CloudFormation tindakan CodePipeline, lihat [referensi konfigurasi tindakan AWS CloudFormation penerapan](#) di Panduan AWS CodePipeline Pengguna.
- Misalnya nilai templat menurut penyedia tindakan, seperti untuk Owner bidang atau configuration bidang, lihat [referensi struktur tindakan](#) di Panduan AWS CodePipeline Pengguna.
- Untuk mengunduh contoh templat tumpukan pipeline dalam format YAMAL atau JSON, lihat [Tutorial: Membuat pipeline dengan AWS CloudFormationAWS CodePipeline](#) Panduan Pengguna.

Menggunakan fungsi penggantian parameter dengan saluran pipa CodePipeline

Dalam satu CodePipeline tahap, Anda dapat menentukan [penggantian parameter untuk AWS CloudFormation tindakan](#). Penggantian parameter memungkinkan Anda menentukan nilai parameter template yang mengesampingkan nilai dalam file konfigurasi templat. AWS CloudFormation menyediakan fungsi untuk membantu Anda menentukan nilai dinamis (nilai yang tidak diketahui hingga pipeline berjalan).

Topik

- [Fn::GetArtifactAtt](#)
- [Fn::GetParam](#)
- [Lihat juga](#)

Fn::GetArtifactAtt

Fungsi `Fn::GetArtifactAtt` mengambil nilai atribut dari artefak input, seperti nama bucket S3 tempat artefak disimpan. Gunakan fungsi ini untuk menentukan atribut artefak, seperti nama file atau nama Amazon S3 bucket.

Saat Anda menjalankan pipeline, CodePipeline menyalin dan menulis file ke penyimpanan artefak pipeline (bucket S3). CodePipeline menghasilkan nama file di toko artefak. Nama file ini tidak diketahui sebelum Anda menjalankan pipeline.

Misalnya, dalam pipeline Anda, Anda mungkin memiliki tahap sumber di mana CodePipeline menyalin kode sumber AWS Lambda fungsi Anda ke penyimpanan artefak. Pada tahap berikutnya, Anda memiliki AWS CloudFormation template yang membuat fungsi Lambda, tetapi AWS CloudFormation membutuhkan nama file untuk membuat fungsi. Anda harus menggunakan fungsi `Fn::GetArtifactAtt` untuk meneruskan bucket S3 dan nama file yang tepat.

Sintaksis

Gunakan sintaks berikut untuk mengambil nilai atribut artefak.

```
{ "Fn::GetArtifactAtt" : [ "artifactName", "attributeName" ] }
```

`artifactName`

Nama artefak input. Anda harus menyatakan artefak ini sebagai input untuk tindakan terkait.

`attributeName`

Nama dari atribut artefak yang ingin Anda ambil nilainya. Untuk detail tentang setiap atribut artefak, lihat bagian Atribut berikut.

Contoh

Penempatan parameter berikut menentukan `ObjectKey` parameter `BucketName` dan dengan mengambil nama bucket S3 dan nama file artefak. `LambdaFunctionSource` Contoh ini

mengasumsikan bahwa CodePipeline menyalin kode sumber fungsi Lambda dan menyimpannya sebagai artefak, misalnya, sebagai bagian dari tahap sumber.

```
{
  "BucketName" : { "Fn::GetArtifactAtt" : ["LambdaFunctionSource", "BucketName"]},
  "ObjectKey" : { "Fn::GetArtifactAtt" : ["LambdaFunctionSource", "ObjectKey"]}
}
```

Atribut

Anda dapat mengambil atribut berikut untuk artefak.

BucketName

Nama bucket S3 tempat data artefak disimpan.

ObjectKey

Nama .zip file yang berisi artefak yang dihasilkan oleh CodePipeline, seperti 1ABCyZZ.zip.

URL

URL Amazon Simple Storage Service (Amazon S3) dari artefak, seperti. `https://s3.us-west-2.amazonaws.com/artifactstorebucket-yivczw8jma0c/test/TemplateSo/1ABCyZZ.zip`

Fn::GetParam

Fungsi `Fn::GetParam` mengembalikan nilai dari pasangan nilai kunci dalam file berformat JSON. File JSON harus disertakan dalam artefak.

Gunakan fungsi ini untuk mengambil nilai output dari AWS CloudFormation tumpukan dan menggunakannya sebagai masukan untuk tindakan lain. Misalnya, jika Anda menentukan nama file keluaran untuk suatu AWS CloudFormation tindakan, CodePipeline menyimpan output dalam file JSON dan kemudian menambahkannya ke file artefak .zip keluaran. Gunakan fungsi `Fn::GetParam` untuk mengambil nilai output dan menggunakannya sebagai input untuk tindakan lain.

Sintaksis

Gunakan sintaks berikut untuk mengambil nilai dari pasangan nilai kunci.

```
{ "Fn::GetParam" : [ "artifactName", "JSONFileName", "keyName" ] }
```

artifactName

Nama artefak, yang harus disertakan sebagai artefak input untuk tindakan terkait.

JSONFileName

Nama file JSON yang ada di dalam artefak.

keyName

Nama dari kunci yang ingin Anda ambil nilainya.

Contoh

Contoh-contoh berikut ini menunjukkan cara menggunakan fungsi `Fn::GetParam` dalam penimpaan parameter.

Sintaksis

Penimpaan parameter berikut menentukan parameter `WebSiteURL` dengan mengambil nilai kunci URL dari file `stack-output.json` yang ada di artefak `WebStackOutput`.

```
{  
  "WebSiteURL" : { "Fn::GetParam" : ["WebStackOutput", "stack-output.json", "URL"] }  
}
```

AWS CloudFormation cuplikan template

Cuplikan AWS CloudFormation template berikut, dari CodePipeline pipeline, menunjukkan cara meneruskan output tumpukan. Snippet ini menunjukkan dua tahap definisi alur. Tahap pertama membuat tumpukan dan menyimpan outputnya dalam file `TestOutput.json` di artefak `StackAOutput`. Nilai-nilai ini ditentukan oleh properti `OutputFileName` dan `OutputArtifacts`.

Nama artefak input sumber untuk tahapan adalah `TemplateSource`. Nama file untuk templat tumpukan adalah `teststackA.yaml`, dan nama file untuk file konfigurasi adalah `test-configuration.json`. Dalam kedua tahap, nilai-nilai ini ditentukan untuk properti `TemplateConfiguration` dan `TemplatePath` seperti yang ditunjukkan:

```
TemplateConfiguration: TemplateSource::test-configuration.json
```



```
TemplatePath: TemplateSource::teststackA.yaml
```

Example Membuat tahapan tumpukan A

```
- 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'
```

Pada tahap berikutnya, stack B menggunakan output dari tumpukan A. Dalam properti `ParameterOverrides`, contoh menggunakan fungsi `Fn::GetParam` untuk menentukan parameter `StackBInputParam`. Nilai yang dihasilkan adalah nilai yang terkait dengan kunci `StackAOutputName`.

Example Membuat tahapan tumpukan B

```
- 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" : ["StackA0Output", "TestOutput.json",
"StackA0OutputName"]}
      }
    InputArtifacts:
      - Name: TemplateSource
      - Name: StackA0Output
    RunOrder: '1'
```

Lihat juga

Sumber daya terkait berikut dapat membantu Anda saat menggunakan parameter ini.

- Untuk informasi selengkapnya tentang parameter CloudFormation tindakan CodePipeline, lihat [referensi konfigurasi tindakan AWS CloudFormation penerapan](#) di Panduan AWS CodePipeline Pengguna.
- Misalnya nilai templat menurut penyedia tindakan, seperti untuk Owner bidang atau configuration bidang, lihat [referensi struktur tindakan](#) di Panduan AWS CodePipeline Pengguna.
- Untuk mengunduh contoh templat tumpukan pipeline dalam format YAMAL atau JSON, lihat [Tutorial: Membuat pipeline dengan AWS CloudFormationAWS CodePipeline](#) Panduan Pengguna.

Panduan Referensi AWS CloudFormation Template

Informasi terperinci tentang sumber daya dan komponen yang tersedia untuk membuat AWS CloudFormation templat sekarang dapat ditemukan di [Panduan Referensi AWS CloudFormation Template](#). Panduan ini berfungsi sebagai referensi teknis untuk template dan melengkapi konten memulai di Panduan CloudFormation Pengguna.

Panduan Referensi AWS CloudFormation Template memberikan rincian tentang:

- Jenis sumber daya dan propertinya
- DeletionPolicy dan atribut tingkat sumber daya lainnya
- Fungsi intrinsik, seperti `Ref` dan `Sub`
- Transformasi (CloudFormation makro)
- Skrip pembantu untuk konfigurasi contoh EC2
- Definisi jenis sumber daya yang dapat dibaca mesin

Panduan ini diperbarui secara berkala untuk mencerminkan:

- Jenis sumber daya baru
- Perubahan properti
- Nilai pengembalian baru
- Pembaruan dokumentasi

Keamanan di AWS CloudFormation

Keamanan cloud di AWS adalah prioritas tertinggi. Sebagai AWS pelanggan, Anda mendapat manfaat dari pusat data dan arsitektur jaringan yang dibangun untuk memenuhi persyaratan organisasi yang paling sensitif terhadap keamanan.

Keamanan adalah tanggung jawab bersama antara Anda AWS dan Anda. [Model tanggung jawab bersama](#) menjelaskan hal ini sebagai keamanan cloud dan keamanan dalam cloud:

- Keamanan cloud — AWS bertanggung jawab untuk melindungi infrastruktur yang menjalankan AWS layanan di AWS Cloud. AWS juga memberi Anda layanan yang dapat Anda gunakan dengan aman. Auditor pihak ketiga secara teratur menguji dan memverifikasi efektivitas keamanan kami sebagai bagian dari [Program AWS Kepatuhan Program AWS Kepatuhan](#). Untuk mempelajari tentang program kepatuhan yang berlaku AWS CloudFormation, lihat [AWS Layanan dalam Lingkup oleh AWS Layanan Program Kepatuhan](#).
- Keamanan di cloud — Tanggung jawab Anda ditentukan oleh AWS layanan yang Anda gunakan. Anda juga bertanggung jawab atas faktor lain, yang mencakup sensitivitas data Anda, persyaratan perusahaan Anda, serta undang-undang dan peraturan yang berlaku.

Dokumentasi ini membantu Anda memahami cara menerapkan model tanggung jawab bersama saat menggunakan AWS CloudFormation. Topik berikut menunjukkan cara mengonfigurasi AWS CloudFormation untuk memenuhi tujuan keamanan dan kepatuhan Anda. Anda juga belajar cara menggunakan AWS layanan lain yang membantu Anda memantau dan mengamankan AWS CloudFormation sumber daya Anda.

Topik

- [Lindungi CloudFormation tumpukan agar tidak dihapus](#)
- [Mencegah pembaruan ke sumber daya tumpukan](#)
- [Perlindungan data di AWS CloudFormation](#)
- [Kontrol CloudFormation akses dengan AWS Identity and Access Management](#)
- [Logging panggilan AWS CloudFormation API dengan AWS CloudTrail](#)
- [Keamanan infrastruktur di AWS CloudFormation](#)
- [Ketahanan di AWS CloudFormation](#)
- [Validasi kepatuhan untuk AWS CloudFormation](#)

- [Konfigurasi dan analisis kerentanan di AWS CloudFormation](#)
- [Praktik terbaik keamanan untuk CloudFormation](#)
- [Akses CloudFormation menggunakan endpoint antarmuka \(\)AWS PrivateLink](#)

Lindungi CloudFormation tumpukan agar tidak dihapus

Anda dapat mencegah tumpukan tidak sengaja dihapus dengan mengaktifkan perlindungan terminasi pada tumpukan. Jika pengguna mencoba untuk menghapus tumpukan dengan perlindungan terminasi diaktifkan, penghapusan gagal dan tumpukan, termasuk statusnya, tetap tidak berubah. Anda dapat mengaktifkan perlindungan terminasi di tumpukan saat membuatnya. Perlindungan terminasi pada tumpukan dinonaktifkan secara default. Anda dapat mengatur perlindungan terminasi pada tumpukan dengan status apa pun kecuali `DELETE_IN_PROGRESS` atau `DELETE_COMPLETE`.

Mengaktifkan atau menonaktifkan perlindungan terminasi pada tumpukan meneruskan pilihan yang sama ke tumpukan bersarang milik tumpukan itu juga. Anda tidak dapat mengaktifkan atau menonaktifkan perlindungan terminasi secara langsung di tumpukan nest. Jika pengguna mencoba untuk langsung menghapus tumpukan nest milik tumpukan yang mengaktifkan perlindungan terminasi, operasi gagal dan tumpukan nest tetap tidak berubah.

Namun, jika pengguna melakukan pembaruan tumpukan yang akan menghapus tumpukan nest, AWS CloudFormation menghapus tumpukan nest.

Perlindungan terminasi berbeda dengan menonaktifkan rollback. Perlindungan terminasi hanya berlaku untuk upaya menghapus tumpukan, sementara menonaktifkan rollback berlaku untuk rollback otomatis ketika pembuatan tumpukan gagal.

Untuk mengaktifkan perlindungan terminasi saat membuat tumpukan

Pada halaman Tentukan opsi tumpukan di wizard Buat tumpukan, di bawah Opsi lanjutan, perluas bagian Perlindungan terminasi dan pilih Aktifkan. Untuk informasi selengkapnya, lihat [Konfigurasi opsi tumpukan](#).

Untuk mengaktifkan atau menonaktifkan perlindungan terminasi di tumpukan yang ada

1. Masuk ke AWS Management Console dan buka AWS CloudFormation konsol di <https://console.aws.amazon.com/cloudformation>.
2. Pada bilah navigasi di bagian atas layar, pilih Anda Wilayah AWS.

3. Pilih tumpukan yang Anda inginkan.

Note

Jika NESTED ditampilkan di sebelah nama tumpukan, tumpukan adalah tumpukan nest. Anda hanya dapat mengubah perlindungan terminasi pada tumpukan akar yang dimiliki tumpukan nest.

4. Di panel detail tumpukan, pilih Tindakan tumpukan dan kemudian Edit perlindungan terminasi.

CloudFormation menampilkan kotak dialog Edit perlindungan terminasi.

5. Pilih Aktifkan atau Nonaktifkan, dan kemudian pilih Simpan.

Untuk mengaktifkan atau menonaktifkan perlindungan terminasi pada tumpukan nest

Jika NESTED ditampilkan di sebelah nama tumpukan, tumpukan adalah tumpukan nest. Anda hanya dapat mengubah perlindungan terminasi pada tumpukan akar yang dimiliki tumpukan nest. Untuk mengubah perlindungan terminasi pada tumpukan akar:

1. Masuk ke AWS Management Console dan buka AWS CloudFormation konsol di <https://console.aws.amazon.com/cloudformation>.
2. Pada bilah navigasi di bagian atas layar, pilih Anda Wilayah AWS.
3. Pilih tumpukan nest yang Anda inginkan.
4. Di panel Info tumpukan, di bagian Gambaran Umum, pilih nama tumpukan terdaftar sebagai Tumpukan akar.

CloudFormation menampilkan detail tumpukan untuk tumpukan root.

5. Pilih Tindakan tumpukan lalu pilih Edit Perlindungan Terminasi.

CloudFormation menampilkan kotak dialog Edit perlindungan terminasi.

6. Pilih Aktifkan atau Nonaktifkan, dan kemudian pilih Simpan.

Untuk mengaktifkan atau menonaktifkan perlindungan pengakhiran menggunakan baris perintah

Gunakan [update-termination-protection](#) perintah.

Mengendalikan siapa yang dapat mengubah perlindungan terminasi pada tumpukan

Untuk mengaktifkan atau menonaktifkan perlindungan terminasi di tumpukan, pengguna memerlukan izin ke tindakan `cloudformation:UpdateTerminationProtection`. Misalnya, kebijakan di bawah ini mengizinkan pengguna untuk mengaktifkan atau menonaktifkan perlindungan terminasi pada tumpukan.

Untuk informasi selengkapnya tentang menentukan izin di AWS CloudFormation, lihat. [Kontrol CloudFormation akses dengan AWS Identity and Access Management](#)

Example Contoh kebijakan yang memberikan izin untuk mengubah perlindungan terminasi tumpukan

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "cloudformation:UpdateTerminationProtection"
    ],
    "Resource": "*"
  }]
}
```

Mencegah pembaruan ke sumber daya tumpukan

Saat Anda membuat tumpukan, semua tindakan pembaruan diizinkan di semua sumber daya. Secara default, siapa pun yang memiliki izin pembaruan tumpukan dapat memperbarui semua sumber daya di tumpukan. Selama pembaruan, beberapa sumber daya mungkin memerlukan gangguan atau diganti sepenuhnya, menghasilkan penyimpanan fisik IDs atau yang sama sekali baru. Anda dapat mencegah sumber daya tumpukan tidak sengaja diperbarui atau dihapus selama pembaruan tumpukan dengan menggunakan kebijakan tumpukan. Kebijakan tumpukan adalah dokumen JSON yang mendefinisikan tindakan pembaruan yang dapat dilakukan pada sumber daya yang ditentukan.

Setelah Anda menetapkan kebijakan tumpukan, semua sumber daya di tumpukan dilindungi secara default. Untuk mengizinkan pembaruan pada sumber daya tertentu, Anda menentukan pernyataan eksplisit `Allow` untuk sumber daya tersebut dalam kebijakan tumpukan Anda. Anda hanya dapat menentukan satu kebijakan tumpukan per tumpukan, tetapi, Anda dapat melindungi

beberapa sumber daya dalam satu kebijakan. Kebijakan tumpukan berlaku untuk semua pengguna CloudFormation yang mencoba untuk memperbarui tumpukan. Anda tidak dapat mengaitkan kebijakan tumpukan yang berbeda dengan pengguna yang berbeda.

Kebijakan tumpukan berlaku hanya selama pembaruan tumpukan. Itu tidak menyediakan kontrol akses seperti kebijakan AWS Identity and Access Management (IAM). Gunakan kebijakan tumpukan hanya sebagai mekanisme gagal-aman untuk mencegah pembaruan yang tidak disengaja untuk sumber daya tumpukan tertentu. Untuk mengontrol akses ke AWS sumber daya atau tindakan, gunakan IAM.

Topik

- [Contoh kebijakan tumpukan](#)
- [Menetapkan kebijakan tumpukan](#)
- [Mengatur kebijakan tumpukan](#)
- [Memperbarui sumber daya dilindungi](#)
- [Mengubah kebijakan tumpukan](#)
- [Lebih banyak contoh kebijakan tumpukan](#)

Contoh kebijakan tumpukan

Kebijakan tumpukan contoh berikut mencegah pembaruan ke sumber daya ProductionDatabase:

```
{
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "*"
    },
    {
      "Effect" : "Deny",
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "LogicalResourceId/ProductionDatabase"
    }
  ]
}
```


Saat Anda menetapkan kebijakan tumpukan, semua sumber daya dilindungi secara default. Untuk mengizinkan pembaruan pada semua sumber daya, kami menambahkan pernyataan Allow yang mengizinkan semua tindakan pada semua sumber daya. Meskipun pernyataan Allow menentukan semua sumber daya, pernyataan eksplisit Deny menyimpannya untuk sumber daya dengan ID logis `ProductionDatabase`. Pernyataan Deny ini mencegah semua tindakan pembaruan, seperti penggantian atau penghapusan, pada sumber daya `ProductionDatabase`.

Elemen `Principal` diperlukan, tetapi hanya mendukung wild card (*), yang berarti bahwa pernyataan tersebut berlaku untuk semua [yang utama](#).

Note

Selama pembaruan tumpukan, CloudFormation secara otomatis memperbarui sumber daya yang bergantung pada sumber daya yang diperbarui lainnya. Misalnya, CloudFormation memperbarui sumber daya yang mereferensikan sumber daya yang diperbarui.

CloudFormation Tidak membuat perubahan fisik, seperti ID sumber daya, untuk memperbarui sumber daya secara otomatis, tetapi jika kebijakan tumpukan dikaitkan dengan sumber daya tersebut, Anda harus memiliki izin untuk memperbaruinya.

Menetapkan kebijakan tumpukan

Saat Anda membuat tumpukan, tidak ada kebijakan tumpukan yang diatur, jadi semua tindakan pembaruan diizinkan di semua sumber daya. Untuk melindungi sumber daya tumpukan dari tindakan pembaruan, tentukan kebijakan tumpukan, lalu tetapkan di tumpukan Anda. Kebijakan tumpukan adalah dokumen JSON yang mendefinisikan tindakan pembaruan CloudFormation tumpukan yang dapat dilakukan CloudFormation pengguna dan sumber daya yang diterapkan tindakan tersebut. Anda mengatur kebijakan tumpukan saat Anda membuat tumpukan, dengan menentukan file teks yang berisi kebijakan tumpukan Anda atau mengetiknya. Saat Anda menetapkan kebijakan tumpukan di tumpukan Anda, pembaruan apa pun yang tidak diizinkan secara eksplisit akan ditolak secara default.

Anda menentukan kebijakan tumpukan dengan lima elemen: `Effect`, `Action`, `Principal`, `Resource`, dan `Condition`. Kode semu berikut menunjukkan sintaks kebijakan tumpukan.

```
{
  "Statement" : [
    {
```

```

    "Effect" : "Deny_or-Allow",
    "Action" : "update_actions",
    "Principal" : "*",
    "Resource" : "LogicalResourceId/resource_logical_ID",
    "Condition" : {
        "StringEquals_or_StringLike" : {
            "ResourceType" : [resource_type, ...]
        }
    }
}
]
}

```

Effect

Menentukan apakah tindakan yang Anda tentukan ditolak atau diizinkan pada sumber daya yang Anda tentukan. Anda hanya dapat menentukan Deny atau Allow, seperti:

```
"Effect" : "Deny"
```

Important

Jika kebijakan tumpukan menyertakan pernyataan yang tumpang tindih (mengizinkan dan menolak pembaruan pada sumber daya), pernyataan Deny selalu menimpa pernyataan Allow. Untuk memastikan bahwa sumber daya dilindungi, gunakan pernyataan Deny untuk sumber daya tersebut.

Tindakan

Menentukan tindakan pembaruan yang ditolak atau diizinkan:

Pembaruan:Ubah

Menentukan tindakan pembaruan selama sumber daya mungkin tidak mengalami gangguan atau beberapa gangguan saat perubahan diterapkan. Semua sumber daya mempertahankan fisik mereka IDs.

Pembaruan:Ganti

Menentukan tindakan pembaruan selama sumber daya diciptakan kembali. CloudFormation menciptakan sumber daya baru dengan pembaruan tertentu dan kemudian menghapus

sumber daya lama. Karena sumber daya diciptakan kembali, ID fisik sumber daya baru mungkin berbeda.

Pembaruan:Hapus

Menentukan tindakan pembaruan selama sumber daya yang dihapus. Pembaruan yang benar-benar menghapus sumber daya dari template tumpukan memerlukan tindakan ini.

Perbarui:*

Menentukan semua tindakan pembaruan. Tanda bintang adalah wild card yang mewakili semua tindakan pembaruan.

Contoh berikut menunjukkan cara menentukan hanya mengganti dan menghapus tindakan:

```
"Action" : ["Update:Replace", "Update:Delete"]
```

Untuk mengizinkan semua tindakan pembaruan kecuali satu, gunakan `NotAction`. Misalnya, untuk mengizinkan semua tindakan pembaruan kecuali untuk `Update:Delete`, gunakan `NotAction`, seperti yang ditunjukkan dalam contoh ini:

```
{
  "Statement" : [
    {
      "Effect" : "Allow",
      "NotAction" : "Update:Delete",
      "Principal": "*",
      "Resource" : "*"
    }
  ]
}
```

Utama

Elemen `Principal` menentukan entitas tempat kebijakan tersebut diterapkan. Elemen ini diperlukan, tetapi hanya mendukung wild card (*), yang berarti bahwa pernyataan tersebut berlaku untuk semua yang [utama](#).

Sumber Daya

Menentukan logika IDs sumber daya yang berlaku untuk kebijakan tersebut. Untuk menentukan jenis sumber daya, gunakan `Condition` elemen.

Untuk menentukan sumber daya tunggal, gunakan ID logisnya. Misalnya:

```
"Resource" : ["LogicalResourceId/myEC2instance"]
```

Anda dapat menggunakan kartu liar dengan logis IDs. Misalnya, jika Anda menggunakan prefiks ID logis umum untuk semua sumber daya terkait, Anda dapat menentukan semuanya dengan wild card:

```
"Resource" : ["LogicalResourceId/CriticalResource*"]
```

Anda juga dapat menggunakan elemen Not dengan sumber daya. Misalnya, untuk mengizinkan pembaruan ke semua sumber daya kecuali untuk satu, gunakan elemen NotResource untuk melindungi sumber daya tersebut:

```
{
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : "Update:*",
      "Principal": "*",
      "NotResource" : "LogicalResourceId/ProductionDatabase"
    }
  ]
}
```

Saat Anda menetapkan kebijakan tumpukan, pembaruan apa pun yang tidak diizinkan secara eksplisit akan ditolak. Dengan mengizinkan pembaruan untuk semua sumber daya kecuali untuk sumber daya ProductionDatabase, Anda menolak pembaruan untuk sumber daya ProductionDatabase.

Ketentuan

Menentukan jenis sumber daya tempat kebijakan tersebut diterapkan. Untuk menentukan logika IDs sumber daya tertentu, gunakan Resource elemen.

Anda dapat menentukan jenis sumber daya, seperti semua EC2 dan instans RDS DB, seperti yang ditunjukkan pada contoh berikut:

```
{
  "Statement" : [
    {
```

```
"Effect" : "Deny",
"Principal" : "*",
"Action" : "Update:*",
"Resource" : "*",
"Condition" : {
  "StringEquals" : {
    "ResourceType" : ["AWS::EC2::Instance", "AWS::RDS::DBInstance"]
  }
},
{
  "Effect" : "Allow",
  "Principal" : "*",
  "Action" : "Update:*",
  "Resource" : "*"
}
]
```

AllowPernyataan tersebut memberikan izin pembaruan ke semua sumber daya dan Deny pernyataan tersebut menolak pembaruan dan instans RDS EC2 DB. Pernyataan Deny selalu menimpa tindakan izinkan.

Anda dapat menggunakan wild card dengan jenis sumber daya. Misalnya, Anda dapat menolak izin pembaruan ke semua EC2 sumber daya Amazon—seperti instans, grup keamanan, dan subnet—dengan menggunakan kartu liar, seperti yang ditunjukkan pada contoh berikut:

```
"Condition" : {
  "StringLike" : {
    "ResourceType" : ["AWS::EC2::*"]
  }
}
```

Anda harus menggunakan syarat StringLike ketika Anda menggunakan wild card.

Mengatur kebijakan tumpukan

Anda dapat menggunakan konsol atau AWS CLI menerapkan kebijakan tumpukan saat membuat tumpukan. Anda juga dapat menggunakan AWS CLI untuk menerapkan kebijakan tumpukan ke tumpukan yang ada. Setelah menerapkan kebijakan tumpukan, Anda tidak dapat menghapusnya dari tumpukan, tetapi Anda dapat menggunakannya AWS CLI untuk memodifikasinya.

Kebijakan tumpukan berlaku untuk semua CloudFormation pengguna yang mencoba memperbarui tumpukan. Anda tidak dapat mengaitkan kebijakan tumpukan yang berbeda dengan pengguna yang berbeda.

Untuk informasi selengkapnya tentang kebijakan tumpukan tulis, lihat [Menetapkan kebijakan tumpukan](#).

Untuk mengatur kebijakan tumpukan saat Anda membuat tumpukan (konsol)

1. Buka AWS CloudFormation konsol di <https://console.aws.amazon.com/cloudformation>.
2. Pada bilah navigasi di bagian atas layar, pilih Wilayah AWS untuk membuat tumpukan.
3. Pada halaman CloudFormation Stacks, pilih Create stack.
4. Dalam panduan Membuat Tumpukan, di halaman Mengkonfigurasi opsi tumpukan, perluas bagian Lanjutan dan kemudian pilih Kebijakan tumpukan.
5. Tentukan kebijakan tumpukan:
 - Untuk menulis kebijakan secara langsung di konsol, pilih Masukkan kebijakan tumpukan, kemudian ketikkan kebijakan tumpukan secara langsung di kolom teks.
 - Untuk menggunakan kebijakan yang ditetapkan dalam file terpisah, pilih Unggah file, kemudian Pilih file untuk memilih file yang berisi kebijakan tumpukan.

Untuk mengatur kebijakan tumpukan saat Anda membuat tumpukan (AWS CLI)

- Gunakan perintah [create-stack](#) dengan opsi `--stack-policy-body` untuk mengetik dalam kebijakan yang dimodifikasi atau opsi `--stack-policy-url` untuk menentukan file yang berisi kebijakan.

Untuk menetapkan kebijakan tumpukan pada tumpukan yang ada (AWS CLI hanya)

- Gunakan perintah [set-stack-policy](#) dengan opsi `--stack-policy-body` untuk mengetik dalam kebijakan yang dimodifikasi atau opsi `--stack-policy-url` untuk menentukan file yang berisi kebijakan.

 Note

Untuk menambahkan kebijakan ke tumpukan yang ada, Anda harus memiliki izin untuk CloudFormation [SetStackPolicy](#) tindakan tersebut.

Memperbarui sumber daya dilindungi

Untuk memperbarui sumber daya yang dilindungi, buat kebijakan sementara yang menimpa kebijakan tumpukan dan memungkinkan pembaruan pada sumber daya tersebut. Tentukan kebijakan timpa saat Anda memperbarui tumpukan. Kebijakan timpa tidak secara permanen mengubah kebijakan tumpukan.

Untuk memperbarui sumber daya yang dilindungi, Anda harus memiliki izin untuk menggunakan CloudFormation [SetStackPolicy](#) tindakan. Untuk informasi tentang mengatur izin CloudFormation, lihat [Kontrol CloudFormation akses dengan AWS Identity and Access Management](#).

Note

Selama pembaruan tumpukan, CloudFormation secara otomatis memperbarui sumber daya yang bergantung pada sumber daya yang diperbarui lainnya. Misalnya, CloudFormation memperbarui sumber daya yang mereferensikan sumber daya yang diperbarui.

CloudFormation Tidak membuat perubahan fisik, seperti ID sumber daya, untuk memperbarui sumber daya secara otomatis, tetapi jika kebijakan tumpukan dikaitkan dengan sumber daya tersebut, Anda harus memiliki izin untuk memperbaruinya.

Untuk memperbarui sumber daya yang dilindungi (konsol)

1. Buka AWS CloudFormation konsol di <https://console.aws.amazon.com/cloudformation>.
2. Pilih tumpukan yang ingin Anda perbarui, pilih Tindakan tumpukan, lalu pilih Perbarui tumpukan.
3. Jika Anda belum mengubah tumpukan templat, pilih Gunakan templat saat ini, lalu klik Selanjutnya. Jika Anda telah mengubah templat, pilih Ganti templat saat ini dan tentukan lokasi templat yang diperbarui di bagian Tentukan templat:
 - Untuk templat yang disimpan secara lokal di komputer Anda, pilih Unggah file templat. Pilih File untuk menavigasi ke file dan memilihnya, dan kemudian klik Selanjutnya.
 - Untuk template yang disimpan dalam bucket Amazon S3, pilih URL Amazon S3. Masukkan atau tempel URL untuk templat, dan kemudian klik Selanjutnya.

Jika Anda memiliki template dalam bucket berkemampuan versi, Anda dapat menentukan versi tertentu dari template template dengan menambahkan `?versionId=version-id` ke URL. Untuk informasi selengkapnya, lihat [Bekerja dengan objek dalam bucket berkemampuan versi](#) di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

4. Jika templat Anda berisi parameter, di halaman Tentukan detail tumpukan, masukkan atau ubah nilai parameter, lalu pilih Selanjutnya.

CloudFormation mengisi setiap parameter dengan nilai yang saat ini diatur dalam tumpukan kecuali untuk parameter yang dideklarasikan dengan NoEcho atribut. Anda dapat menggunakan nilai saat ini untuk parameter tersebut dengan memilih Gunakan nilai yang ada.

Untuk informasi selengkapnya tentang penggunaan NoEcho untuk menutupi informasi sensitif, serta menggunakan parameter dinamis untuk mengelola rahasia, lihat praktik [Jangan menanamkan kredensial dalam templat Anda](#) terbaik.

5. Tentukan kebijakan tanpa tumpukan.
 - a. Di halaman Mengonfigurasi opsi tumpukan, di bagian Opsi lanjutan, pilih Kebijakan tumpukan .
 - b. Pilih Unggah file.
 - c. Klik Pilih file dan arahkan ke file yang berisi kebijakan tumpukan utama atau ketik kebijakan.
 - d. Pilih Berikutnya.

Kebijakan tanpa harus menentukan sebuah pernyataan Allow untuk sumber daya yang dilindungi yang ingin Anda perbarui. Misalnya, untuk memperbarui semua sumber daya yang dilindungi, tentukan kebijakan tanpa sementara yang memungkinkan semua pembaruan:

```
{
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "*"
    }
  ]
}
```


 Note

CloudFormation menerapkan kebijakan penggantian hanya selama pembaruan ini. Kebijakan tanpa tidak secara permanen mengubah kebijakan tumpukan. Untuk mengubah kebijakan tumpukan, lihat [Mengubah kebijakan tumpukan](#).

6. Tinjau informasi tumpukan dan perubahan yang Anda kirimkan.

Periksa bahwa Anda telah mengirimkan informasi yang benar, seperti nilai parameter atau URL templat yang benar. Jika template Anda berisi sumber daya IAM, pilih Saya mengakui bahwa template ini dapat membuat sumber daya IAM untuk menentukan bahwa Anda ingin menggunakan sumber daya IAM dalam template. Untuk informasi selengkapnya, lihat [Mengakui sumber daya IAM dalam templat CloudFormation](#).

Di bagian Pratinjau perubahan Anda, periksa apakah CloudFormation akan membuat semua perubahan yang Anda harapkan. Misalnya, periksa apakah CloudFormation menambahkan, menghapus, dan memodifikasi sumber daya yang ingin Anda tambahkan, hapus, atau modifikasi. CloudFormation menghasilkan pratinjau ini dengan membuat set perubahan untuk tumpukan. Untuk informasi selengkapnya, lihat [Perbarui CloudFormation tumpukan menggunakan set perubahan](#).

7. Setelah Anda puas dengan perubahan Anda, klik Perbarui.

 Note

Pada titik ini, Anda juga memiliki opsi untuk melihat set perubahan untuk meninjau pembaruan yang diajukan dengan lebih teliti. Untuk melakukannya, klik Lihat set perubahan, bukan Perbarui. CloudFormation menampilkan set perubahan yang dihasilkan berdasarkan pembaruan Anda. Jika Anda siap untuk melakukan pembaruan tumpukan, klik Jalankan.

CloudFormation menampilkan halaman detail Stack untuk tumpukan Anda. Tumpukan Anda sekarang memiliki status `UPDATE_IN_PROGRESS`. Setelah CloudFormation berhasil selesai memperbarui tumpukan, ia menetapkan status tumpukan ke `UPDATE_COMPLETE`.

Jika pembaruan tumpukan gagal, CloudFormation; secara otomatis memutar kembali perubahan, dan menetapkan status tumpukan ke `UPDATE_ROLLBACK_COMPLETE`.

Untuk memperbarui sumber daya yang dilindungi (AWS CLI)

- Gunakan perintah [update-stack](#) dengan opsi `--stack-policy-during-update-body` untuk mengetik dalam kebijakan yang dimodifikasi atau opsi `--stack-policy-during-update-url` untuk menentukan file yang berisi kebijakan.

 Note

CloudFormation menerapkan kebijakan penggantian hanya selama pembaruan ini. Kebijakan tanpa tidak secara permanen mengubah kebijakan tumpukan. Untuk mengubah kebijakan tumpukan, lihat [Mengubah kebijakan tumpukan](#).

Mengubah kebijakan tumpukan

Untuk melindungi sumber daya tambahan atau untuk menghapus perlindungan dari sumber daya, ubah kebijakan tumpukan. Misalnya, saat Anda menambahkan basis data yang ingin Anda lindungi ke tumpukan Anda, tambahkan pernyataan Deny untuk basis data tersebut ke kebijakan tumpukan. Untuk mengubah kebijakan, Anda harus memiliki izin untuk menggunakan tindakan [SetStackPolicy](#).

Gunakan AWS CLI untuk memodifikasi kebijakan tumpukan.

Untuk mengubah kebijakan tumpukan, lihat (AWS CLI)

- Gunakan perintah [set-stack-policy](#) dengan opsi `--stack-policy-body` untuk mengetik dalam kebijakan yang dimodifikasi atau opsi `--stack-policy-url` untuk menentukan file yang berisi kebijakan.

Anda tidak dapat menghapus kebijakan tumpukan. Untuk menghapus semua perlindungan dari semua sumber daya, Anda mengubah kebijakan untuk secara eksplisit mengizinkan semua tindakan pada semua sumber daya. Kebijakan berikut ini mengizinkan semua pembaruan pada semua sumber daya:

```
{
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : "Update:*",
      "Principal": "*",
```

```
    "Resource" : "*"
  }
]
}
```

Lebih banyak contoh kebijakan tumpukan

Contoh kebijakan berikut menunjukkan cara mencegah pembaruan ke semua sumber daya tumpukan dan sumber daya tertentu, serta mencegah jenis pembaruan tertentu.

Mencegah pembaruan ke semua sumber daya tumpukan

Untuk mencegah pembaruan ke semua sumber daya tumpukan, kebijakan berikut menetapkan pernyataan Deny untuk semua tindakan pembaruan pada semua sumber daya.

```
{
  "Statement" : [
    {
      "Effect" : "Deny",
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "*"
    }
  ]
}
```

Mencegah pembaruan ke sumber daya tunggal

Kebijakan berikut menolak semua tindakan pembaruan pada basis data dengan ID logis MyDatabase. Hal ini memungkinkan semua tindakan pembaruan pada semua sumber daya tumpukan lainnya dengan pernyataan Allow. Pernyataan Allow tidak berlaku untuk sumber daya MyDatabase karena pernyataan Deny selalu menimpa tindakan izinkan.

```
{
  "Statement" : [
    {
      "Effect" : "Deny",
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "LogicalResourceId/MyDatabase"
    },
  ],
}
```

```
{
  "Effect" : "Allow",
  "Action" : "Update:*",
  "Principal" : "*",
  "Resource" : "*"
}
```

Anda dapat mencapai hasil yang sama seperti contoh sebelumnya dengan menggunakan penolakan default. Saat Anda menetapkan kebijakan tumpukan, tolak CloudFormation pembaruan apa pun yang tidak diizinkan secara eksplisit. Kebijakan berikut ini mengizinkan pembaruan untuk semua sumber daya kecuali untuk sumber daya `ProductionDatabase`, yang ditolak secara default.

```
{
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : "Update:*",
      "Principal" : "*",
      "NotResource" : "LogicalResourceId/ProductionDatabase"
    }
  ]
}
```

Important

Ada risiko dalam menggunakan penolakan default. Jika Anda memiliki pernyataan `Allow` di bagian lain dalam kebijakan (seperti pernyataan `Allow` yang menggunakan wild card), Anda mungkin secara tidak sadar memberikan izin pembaruan ke sumber daya yang tidak Anda inginkan. Karena penolakan eksplisit menimpa tindakan izinkan, Anda bisa memastikan bahwa sumber daya dilindungi dengan menggunakan pernyataan `Deny`.

Mencegah pembaruan untuk semua instans dari jenis sumber daya

Kebijakan berikut menolak semua tindakan pembaruan pada jenis sumber daya instans DB RDS. Hal ini memungkinkan semua tindakan pembaruan pada semua sumber daya tumpukan lainnya dengan pernyataan `Allow`. Pernyataan `Allow` tidak berlaku untuk sumber daya instans DB RDS karena pernyataan `Deny` selalu menimpa tindakan izinkan.

```
{
  "Statement" : [
    {
      "Effect" : "Deny",
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "*",
      "Condition" : {
        "StringEquals" : {
          "ResourceType" : ["AWS::RDS::DBInstance"]
        }
      }
    },
    {
      "Effect" : "Allow",
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "*"
    }
  ]
}
```

Mencegah pembaruan pengganti untuk sebuah instans

Kebijakan berikut menolak pembaruan yang akan menyebabkan penggantian instans dengan ID logis MyInstance. Hal ini memungkinkan semua tindakan pembaruan pada semua sumber daya tumpukan lainnya dengan pernyataan Allow. Pernyataan Allow tidak berlaku untuk sumber daya MyInstance karena pernyataan Deny selalu menimpa tindakan izinkan.

```
{
  "Statement" : [
    {
      "Effect" : "Deny",
      "Action" : "Update:Replace",
      "Principal": "*",
      "Resource" : "LogicalResourceId/MyInstance"
    },
    {
      "Effect" : "Allow",
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "*"
    }
  ]
}
```

```
}  
]  
}
```

Mencegah pembaruan pada tumpukan nested

Kebijakan berikut menyangkal semua tindakan pembaruan pada jenis sumber daya CloudFormation tumpukan (tumpukan bersarang). Hal ini memungkinkan semua tindakan pembaruan pada semua sumber daya tumpukan lainnya dengan pernyataan Allow. Pernyataan Allow tidak berlaku untuk CloudFormation sumber daya tumpukan karena pernyataan Deny selalu menimpa tindakan izinkan.

```
{  
  "Statement" : [  
    {  
      "Effect" : "Deny",  
      "Action" : "Update:*",  
      "Principal": "*",  
      "Resource" : "*",  
      "Condition" : {  
        "StringEquals" : {  
          "ResourceType" : ["AWS::CloudFormation::Stack"]  
        }  
      }  
    },  
    {  
      "Effect" : "Allow",  
      "Action" : "Update:*",  
      "Principal": "*",  
      "Resource" : "*"   
    }  
  ]  
}
```

Perlindungan data di AWS CloudFormation

[Model tanggung jawab AWS bersama model](#) berlaku untuk perlindungan data di AWS

CloudFormation. Seperti yang dijelaskan dalam model AWS ini, bertanggung jawab untuk melindungi infrastruktur global yang menjalankan semua AWS Cloud. Anda bertanggung jawab untuk mempertahankan kendali atas konten yang di-host pada infrastruktur ini. Anda juga bertanggung jawab atas tugas-tugas konfigurasi dan manajemen keamanan untuk Layanan AWS yang Anda

gunakan. Lihat informasi yang lebih lengkap tentang privasi data dalam [Pertanyaan Umum Privasi Data](#). Lihat informasi tentang perlindungan data di Eropa di pos blog [Model Tanggung Jawab Bersama dan GDPR AWS](#) di Blog Keamanan AWS .

Untuk tujuan perlindungan data, kami menyarankan Anda melindungi Akun AWS kredensial dan mengatur pengguna individu dengan AWS IAM Identity Center atau AWS Identity and Access Management (IAM). Dengan cara itu, setiap pengguna hanya diberi izin yang diperlukan untuk memenuhi tanggung jawab tugasnya. Kami juga menyarankan supaya Anda mengamankan data dengan cara-cara berikut:

- Gunakan autentikasi multi-faktor (MFA) pada setiap akun.
- Gunakan SSL/TLS untuk berkomunikasi dengan sumber daya. AWS Kami mensyaratkan TLS 1.2 dan menganjurkan TLS 1.3.
- Siapkan API dan logging aktivitas pengguna dengan AWS CloudTrail. Untuk informasi tentang penggunaan CloudTrail jejak untuk menangkap AWS aktivitas, lihat [Bekerja dengan CloudTrail jejak](#) di AWS CloudTrail Panduan Pengguna.
- Gunakan solusi AWS enkripsi, bersama dengan semua kontrol keamanan default di dalamnya Layanan AWS.
- Gunakan layanan keamanan terkelola tingkat lanjut seperti Amazon Macie, yang membantu menemukan dan mengamankan data sensitif yang disimpan di Amazon S3.
- Jika Anda memerlukan modul kriptografi tervalidasi FIPS 140-3 saat mengakses AWS melalui antarmuka baris perintah atau API, gunakan titik akhir FIPS. Lihat informasi selengkapnya tentang titik akhir FIPS yang tersedia di [Standar Pemrosesan Informasi Federal \(FIPS\) 140-3](#).

Kami sangat merekomendasikan agar Anda tidak pernah memasukkan informasi identifikasi yang sensitif, seperti nomor rekening pelanggan Anda, ke dalam tanda atau bidang isian bebas seperti bidang Nama. Ini termasuk saat Anda bekerja dengan AWS CloudFormation atau lainnya Layanan AWS menggunakan konsol, API AWS CLI, atau AWS SDKs. Data apa pun yang Anda masukkan ke dalam tanda atau bidang isian bebas yang digunakan untuk nama dapat digunakan untuk log penagihan atau log diagnostik. Saat Anda memberikan URL ke server eksternal, kami sangat menganjurkan supaya Anda tidak menyertakan informasi kredensial di dalam URL untuk memvalidasi permintaan Anda ke server itu.

Enkripsi diam

Mengikuti model tanggung jawab AWS bersama, AWS CloudFormation menyimpan data Anda yang dienkripsi saat istirahat. Pelanggan bertanggung jawab untuk mengatur kebijakan enkripsi

dan penyimpanan untuk data yang disimpan di akun mereka. Sebagai contoh, sebaiknya aktifkan enkripsi yang tidak aktif untuk templat dan data lain yang disimpan dalam Bucket S3 atau topik SNS. Pelanggan juga menentukan pengaturan enkripsi untuk setiap sistem penyimpanan data yang ditetapkan oleh CloudFormation.

Enkripsi bergerak

AWS CloudFormation menggunakan saluran terenkripsi untuk komunikasi layanan di bawah model tanggung jawab bersama.

Privasi lalu lintas antar jaringan

AWS CloudFormation komunikasi layanan dienkripsi secara aman secara default antara Wilayah atau Availability Zone.

Kontrol CloudFormation akses dengan AWS Identity and Access Management

Dengan AWS Identity and Access Management (IAM), Anda dapat membuat pengguna IAM dan mengontrol akses mereka ke sumber daya tertentu di Anda. Akun AWS Saat Anda menggunakan IAM, Anda dapat mengontrol apa yang dapat dilakukan pengguna CloudFormation, seperti apakah mereka dapat melihat templat tumpukan, membuat tumpukan, atau menghapus tumpukan.

Di luar tindakan CloudFormation spesifik, Anda dapat mengelola AWS layanan dan sumber daya apa yang tersedia untuk setiap pengguna. Dengan begitu, Anda dapat mengontrol sumber daya mana yang dapat diakses pengguna saat mereka menggunakannya CloudFormation. Misalnya, Anda dapat menentukan pengguna mana yang dapat membuat EC2 instance Amazon, menghentikan instance database, atau memperbarui VPCs Izin yang sama berlaku kapan saja mereka gunakan CloudFormation untuk melakukan tindakan tersebut.

Gunakan informasi di bagian berikut untuk mengontrol siapa yang dapat mengakses CloudFormation. Kami juga akan mengeksplorasi cara mengotorisasi pembuatan sumber daya IAM dalam template, memberikan aplikasi yang berjalan pada EC2 instance izin yang mereka butuhkan, dan memanfaatkan kredensial keamanan sementara untuk meningkatkan keamanan di lingkungan Anda.

AWS

Mendefinisikan kebijakan berbasis identitas IAM untuk CloudFormation

Untuk memberikan akses ke CloudFormation, Anda perlu membuat dan menetapkan kebijakan IAM yang memberikan identitas IAM Anda (seperti pengguna atau peran) izin untuk memanggil tindakan API yang mereka butuhkan.

Dengan kebijakan berbasis identitas IAM, Anda dapat menentukan tindakan dan sumber daya yang diizinkan atau ditolak, serta kondisi di mana tindakan diizinkan atau ditolak. CloudFormation mendukung tindakan, sumber daya, dan kunci kondisi tertentu.

Jika Anda baru mengenal IAM, mulailah dengan membiasakan diri dengan elemen-elemen kebijakan IAM JSON. Untuk informasi selengkapnya, lihat [referensi elemen kebijakan IAM JSON](#) di Panduan Pengguna IAM. Untuk mempelajari cara membuat kebijakan IAM, selesaikan tutorial [Buat dan lampirkan kebijakan terkelola pelanggan pertama Anda](#) dalam dokumentasi IAM.

Topik

- [Tindakan kebijakan untuk CloudFormation](#)
- [Tindakan khusus konsol untuk CloudFormation](#)
- [Sumber daya kebijakan untuk CloudFormation](#)
- [Kunci kondisi kebijakan untuk CloudFormation](#)

Tindakan kebijakan untuk CloudFormation

Dalam Action elemen pernyataan kebijakan IAM Anda, Anda dapat menentukan tindakan API apa pun yang CloudFormation ditawarkan. Anda harus mengawali nama tindakan dengan string huruf kecil. `cloudformation:` Misalnya: `cloudformation:CreateStack`, `cloudformation:CreateChangeSet`, dan `cloudformation:UpdateStack`.

Untuk menentukan beberapa tindakan dalam satu pernyataan, pisahkan dengan koma, sebagai berikut:

```
"Action": [ "cloudformation:action1", "cloudformation:action2" ]
```

Anda juga dapat menentukan beberapa tindakan menggunakan wildcard. Misalnya, Anda dapat menentukan semua tindakan yang namanya dimulai dengan kata `Get`, sebagai berikut:

```
"Action": "cloudformation:Get*"
```

Untuk melihat daftar lengkap tindakan yang terkait dengan awalan `cloudformation` layanan, lihat [Tindakan, sumber daya, dan kunci kondisi untuk AWS CloudFormation serta Kunci Tindakan, sumber daya, dan kondisi AWS Cloud Control API di Referensi](#) Otorisasi Layanan.

Contoh

Berikut ini menunjukkan contoh kebijakan izin yang memberikan izin untuk melihat tumpukan CloudFormation

Example 1: Contoh kebijakan yang memberikan izin tumpukan tampilan

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "cloudformation:DescribeStacks",
      "cloudformation:DescribeStackEvents",
      "cloudformation:DescribeStackResource",
      "cloudformation:DescribeStackResources"
    ],
    "Resource": "*"
  }]
}
```

Pengguna yang membuat atau menghapus tumpukan memerlukan izin tambahan berdasarkan templat tumpukan mereka. Misalnya, jika template Anda menjelaskan antrean Amazon SQS, pengguna harus memiliki izin untuk tindakan Amazon SQS dan Amazon CloudFormation SQS, seperti yang ditunjukkan dalam kebijakan contoh berikut.

Example 2: Contoh kebijakan yang memberikan tindakan pembuatan dan tampilan tumpukan dan semua tindakan Amazon SQS

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

```
        "sqs:*",
        "cloudformation:CreateStack",
        "cloudformation:DescribeStacks",
        "cloudformation:DescribeStackEvents",
        "cloudformation:DescribeStackResources",
        "cloudformation:GetTemplate",
        "cloudformation:ValidateTemplate"
    ],
    "Resource": "*"
}]
}
```

Tindakan khusus konsol untuk CloudFormation

Pengguna CloudFormation konsol memerlukan izin tambahan di luar yang diperlukan untuk AWS Command Line Interface atau CloudFormation APIs. Izin tambahan ini mendukung fitur khusus konsol seperti unggahan templat ke bucket Amazon S3 dan daftar drop-down untuk tipe parameter khusus. AWS

Untuk semua tindakan yang tercantum di bawah ini, berikan izin ke semua sumber daya; jangan batasi pada tumpukan atau bucket tertentu.

Tindakan berikut hanya digunakan oleh CloudFormation konsol dan tidak didokumentasikan dalam referensi API. Tindakan ini memungkinkan pengguna untuk mengunggah template ke bucket Amazon S3.

- `cloudformation:CreateUploadBucket`

Saat pengguna mengunggah templat, pengguna juga memerlukan izin Amazon S3 berikut:

- `s3:PutObject`
- `s3:ListBucket`
- `s3:GetObject`
- `s3:CreateBucket`

Untuk melihat nilai dalam daftar drop-down parameter untuk template dengan tipe parameter AWS-specific, pengguna memerlukan izin untuk membuat panggilan API describe yang sesuai. Misalnya, izin berikut diperlukan saat jenis parameter ini digunakan dalam templat:

- `ec2:DescribeKeyPairs`— Diperlukan untuk tipe `AWS::EC2::KeyPair::KeyName` parameter.
- `ec2:DescribeSecurityGroups`— Diperlukan untuk tipe `AWS::EC2::SecurityGroup::Id` parameter.
- `ec2:DescribeSubnets`— Diperlukan untuk tipe `AWS::EC2::Subnet::Id` parameter.
- `ec2:DescribeVpcs`— Diperlukan untuk tipe `AWS::EC2::VPC::Id` parameter.

Untuk informasi selengkapnya tentang tipe parameter AWS-spesifik, lihat [Tentukan sumber daya yang ada saat runtime dengan CloudFormation tipe parameter yang disediakan](#).

Sumber daya kebijakan untuk CloudFormation

Dalam pernyataan kebijakan IAM, `Resource` elemen menentukan objek atau objek yang dicakup oleh pernyataan tersebut. Untuk CloudFormation, setiap pernyataan kebijakan IAM berlaku untuk sumber daya yang Anda tentukan menggunakan Amazon Resource Names (ARNs). Format ARN spesifik tergantung pada sumber daya.

Untuk daftar lengkap jenis sumber daya dan jenis CloudFormation sumber daya ARNs, lihat [Jenis sumber daya yang ditentukan oleh AWS CloudFormation](#) dalam Referensi Otorisasi Layanan. Untuk mempelajari tindakan mana yang dapat Anda tentukan dengan ARN masing-masing sumber daya, lihat [Tindakan yang ditentukan oleh](#). AWS CloudFormation

Anda dapat menentukan tindakan untuk tumpukan tertentu, seperti yang ditunjukkan dalam contoh kebijakan berikut. Saat Anda memberikan ARN, ganti *placeholder text* dengan informasi khusus sumber daya Anda.

Example 1: Contoh kebijakan yang menolak tindakan menghapus dan memperbarui tumpukan untuk tumpukan yang ditentukan

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Deny",
    "Action": [
      "cloudformation:DeleteStack",
      "cloudformation:UpdateStack"
    ],
    "Resource": "arn:aws:cloudformation:region:account-id:stack/MyProductionStack/*"
  }]
}
```

```
}
```

Kebijakan di atas menggunakan kartu liar di akhir nama tumpukan sehingga tumpukan hapus dan tumpukan pembaruan ditolak pada ID tumpukan lengkap (seperti `arn:aws:cloudformation:region:account-id:stack/MyProductionStack/abc9dbf0-43c2-11e3-a6e8-50fa526be49c`) dan nama tumpukan (seperti `MyProductionStack`).

Untuk mengizinkan `AWS::Serverless` transformasi membuat set perubahan, sertakan izin `arn:aws:cloudformation:region:aws:transform/Serverless-2016-10-31` tingkat sumber daya, seperti yang ditunjukkan dalam kebijakan berikut.

Example 2: Contoh kebijakan yang memungkinkan tindakan set perubahan buat untuk transformasi yang ditentukan

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "cloudformation:CreateChangeSet"
    ],
    "Resource": "arn:aws:cloudformation:region:aws:transform/Serverless-2016-10-31"
  }]
}
```

Kunci kondisi kebijakan untuk CloudFormation

Dalam pernyataan kebijakan IAM, Anda dapat secara opsional menentukan kondisi yang mengontrol kapan kebijakan berlaku. Misalnya, Anda dapat menentukan kebijakan yang mengizinkan pengguna membuat tumpukan hanya saat mereka menentukan URL templat tertentu. Anda dapat menentukan kondisi CloudFormation -spesifik dan kondisi AWS-wide, seperti `DateLessThan`, yang menentukan kapan kebijakan berhenti berlaku. Untuk informasi selengkapnya dan daftar kondisi AWS-wide, lihat Kondisi dalam [referensi elemen kebijakan IAM](#) di Panduan Pengguna IAM.

Note

Jangan gunakan kondisi `aws:SourceIp` AWS-wide. CloudFormation menyediakan sumber daya dengan menggunakan alamat IP sendiri, bukan alamat IP dari permintaan asal.

Misalnya, saat Anda membuat tumpukan, CloudFormation membuat permintaan dari alamat IP-nya untuk meluncurkan EC2 instans Amazon atau membuat bucket Amazon S3, bukan dari alamat IP dari CreateStack panggilan atau perintah. `create-stack`

Daftar berikut menjelaskan kondisi CloudFormation -spesifik. Syarat ini diterapkan hanya ketika pengguna membuat atau memperbarui tumpukan:

`cloudformation:ChangeSetName`

Nama set CloudFormation perubahan yang ingin Anda kaitkan dengan kebijakan. Gunakan syarat ini untuk mengontrol set perubahan mana yang dapat dijalankan atau dihapus oleh pengguna .

`cloudformation:ImportResourceTypes`

Jenis sumber daya templat yang ingin Anda kaitkan dengan kebijakan, seperti `AWS::EC2::Instance`. Gunakan syarat ini untuk mengontrol jenis sumber daya mana yang dapat digunakan pengguna saat mereka mengimpor sumber daya ke dalam tumpukan. Kondisi ini diperiksa terhadap jenis sumber daya yang dideklarasikan pengguna dalam `ResourcesToImport` parameter, yang saat ini hanya didukung untuk AWS CLI dan permintaan API. Saat menggunakan parameter ini, Anda harus menentukan semua jenis sumber daya yang Anda inginkan agar dikontrol pengguna selama operasi impor. Untuk informasi selengkapnya tentang `ResourcesToImport` parameter, lihat [CreateChangeSet](#) tindakan di Referensi AWS CloudFormation API.

Untuk daftar kemungkinan `ResourcesToImport`, lihat [Dukungan jenis sumber daya](#).

Gunakan konvensi penamaan sumber daya tiga bagian untuk menentukan jenis sumber daya mana yang dapat digunakan pengguna, dari semua sumber daya di seluruh organisasi, hingga jenis sumber daya individual.

`organization::*`

Tentukan semua jenis sumber daya untuk organisasi tertentu.

`organization::service_name::*`

Tentukan semua jenis sumber daya untuk layanan tertentu dalam organisasi tertentu.

`organization::service_name::resource_type`

Tentukan jenis sumber daya tertentu.

Misalnya:

`AWS::*`

Tentukan semua jenis AWS sumber daya yang didukung.

`AWS::service_name::*`

Tentukan semua sumber daya yang didukung untuk layanan AWS tertentu.

`AWS::service_name::resource_type`

Tentukan jenis AWS sumber daya tertentu, seperti `AWS::EC2::Instance` (semua EC2 contoh).

cloudformation:ResourceTypes

Jenis sumber daya templat, seperti `AWS::EC2::Instance`, yang ingin Anda kaitkan dengan kebijakan. Gunakan syarat ini untuk mengontrol jenis sumber daya mana yang dapat digunakan pengguna saat mereka membuat atau memperbarui tumpukan. Kondisi ini diperiksa terhadap jenis sumber daya yang dideklarasikan pengguna dalam `ResourceTypes` parameter, yang saat ini hanya didukung untuk AWS CLI dan permintaan API. Saat menggunakan parameter ini, pengguna harus menentukan semua jenis sumber daya yang ada di templat mereka. Untuk informasi selengkapnya tentang `ResourceTypes` parameter, lihat [CreateStack](#) tindakan di Referensi AWS CloudFormation API.

Untuk daftar jenis sumber daya, lihat [Panduan Referensi AWS CloudFormation Template](#).

Gunakan konvensi penamaan sumber daya tiga bagian untuk menentukan jenis sumber daya mana yang dapat digunakan pengguna, dari semua sumber daya di seluruh organisasi, hingga jenis sumber daya individual.

`organization::*`

Tentukan semua jenis sumber daya untuk organisasi tertentu.

`organization::service_name::*`

Tentukan semua jenis sumber daya untuk layanan tertentu dalam organisasi tertentu.

`organization::service_name::resource_type`

Tentukan jenis sumber daya tertentu.

Misalnya:

`AWS::*`

Tentukan semua jenis AWS sumber daya yang didukung.

`AWS::service_name::*`

Tentukan semua sumber daya yang didukung untuk layanan AWS tertentu.

`AWS::service_name::resource_type`

Tentukan jenis AWS sumber daya tertentu, seperti `AWS::EC2::Instance` (semua EC2 contoh).

`Alexa::ASK::*`

Tentukan semua jenis sumber daya di Alexa Skill Kit.

`Alexa::ASK::Skill`

Tentukan tipe sumber daya [Alexa::ASK::Skill](#) individu.

`Custom::*`

Tentukan semua sumber daya kustom.

Untuk informasi selengkapnya, lihat [Buat logika penyediaan khusus dengan sumber daya khusus](#).

`Custom::resource_type`

Tentukan jenis sumber daya kustom tertentu.

Untuk informasi selengkapnya, lihat [Buat logika penyediaan khusus dengan sumber daya khusus](#).

`cloudformation:RoleARN`

Nama Sumber Daya Amazon (ARN) dari peran layanan IAM yang ingin Anda kaitkan dengan kebijakan. Gunakan syarat ini untuk mengontrol peran layanan mana yang dapat digunakan pengguna saat mereka bekerja dengan tumpukan atau set perubahan.

`cloudformation:StackPolicyUrl`

URL kebijakan tumpukan Amazon S3 yang ingin Anda kaitkan dengan kebijakan. Gunakan syarat ini untuk mengontrol kebijakan tumpukan mana yang dapat dikaitkan pengguna dengan tumpukan

selama tindakan membuat atau memperbarui tumpukan. Untuk informasi selengkapnya tentang kebijakan tumpukan, lihat [Mencegah pembaruan ke sumber daya tumpukan](#).

 Note

Untuk memastikan bahwa pengguna hanya dapat membuat atau memperbarui tumpukan dengan kebijakan tumpukan yang Anda unggah, setel bucket S3 agar hanya dibaca bagi pengguna tersebut.

`cloudformation:TemplateUrl`

URL template Amazon S3 yang ingin Anda kaitkan dengan kebijakan. Gunakan syarat ini untuk mengontrol templat mana yang dapat digunakan pengguna saat mereka membuat atau memperbarui tumpukan.

 Note

Untuk memastikan bahwa pengguna hanya dapat membuat atau memperbarui tumpukan dengan templat yang Anda unggah, setel bucket S3 agar hanya dibaca bagi pengguna tersebut.

 Note

Kondisi CloudFormation spesifik berikut berlaku untuk parameter API dengan nama yang sama:

- `cloudformation:ChangeSetName`
- `cloudformation:RoleARN`
- `cloudformation:StackPolicyUrl`
- `cloudformation:TemplateUrl`

Misalnya, `cloudformation:TemplateUrl` hanya berlaku untuk `TemplateUrl` parameter untuk `CreateStack`, `UpdateStack`, dan `CreateChangeSet` APIs.

Untuk contoh kebijakan IAM yang menggunakan tombol kondisi untuk mengontrol akses, lihat [Contoh kebijakan berbasis identitas IAM untuk CloudFormation](#).

Mengakui sumber daya IAM dalam templat CloudFormation

Sebelum Anda dapat membuat tumpukan, CloudFormation memvalidasi template Anda. Selama validasi, CloudFormation memeriksa template Anda untuk sumber daya IAM yang mungkin dibuatnya. Sumber daya IAM, seperti pengguna dengan akses penuh, dapat mengakses dan memodifikasi sumber daya apa pun di sumber daya Anda Akun AWS. Oleh karena itu, kami menyarankan agar Anda meninjau izin yang terkait dengan setiap sumber daya IAM sebelum melanjutkan sehingga Anda tidak secara tidak sengaja membuat sumber daya dengan izin yang meningkat. Untuk memastikan bahwa Anda telah melakukannya, Anda harus mengakui bahwa template berisi sumber daya tersebut, memberikan kemampuan CloudFormation yang ditentukan sebelum membuat tumpukan.

Anda dapat mengetahui kemampuan CloudFormation template dengan menggunakan CloudFormation konsol, AWS Command Line Interface (AWS CLI), atau API:

- Di CloudFormation konsol, pada halaman Configure stack options dari wizard Create Stack atau Update Stack, pilih Saya mengakui bahwa template ini dapat membuat sumber daya IAM.
- Dalam AWS CLI, ketika Anda menggunakan [update-stack](#) perintah [create-stack](#) dan, tentukan `CAPABILITY_IAM` atau `CAPABILITY_NAMED_IAM` nilai untuk `--capabilities` opsi tersebut. Jika template Anda menyertakan sumber daya IAM, Anda dapat menentukan salah satu kemampuan. Jika template Anda menyertakan nama kustom untuk sumber daya IAM, Anda harus menentukan `CAPABILITY_NAMED_IAM`.
- Di API, saat Anda menggunakan [CreateStack](#) dan [UpdateStack](#) tindakan, tentukan `Capabilities.member.1=CAPABILITY_IAM` atau `Capabilities.member.1=CAPABILITY_NAMED_IAM`. Jika template Anda menyertakan sumber daya IAM, Anda dapat menentukan salah satu kemampuan. Jika template Anda menyertakan nama kustom untuk sumber daya IAM, Anda harus menentukan `CAPABILITY_NAMED_IAM`.

Important

Jika template Anda berisi sumber daya IAM bernama kustom, jangan membuat beberapa tumpukan menggunakan kembali template yang sama. Sumber daya IAM harus unik secara global dalam akun Anda. Jika Anda menggunakan template yang sama untuk membuat beberapa tumpukan di Wilayah yang berbeda, tumpukan Anda mungkin berbagi sumber daya IAM yang sama, daripada masing-masing memiliki yang unik. Sumber daya bersama di antara tumpukan dapat memiliki konsekuensi yang tidak diinginkan yang tidak dapat Anda

pulihkan. Misalnya, jika Anda menghapus atau memperbarui sumber daya IAM bersama dalam satu tumpukan, Anda akan secara tidak sengaja mengubah sumber daya tumpukan lain.

Mengelola kredensi untuk aplikasi yang berjalan di instans Amazon EC2

Jika Anda memiliki aplikasi yang berjalan pada EC2 instans Amazon dan perlu membuat permintaan ke AWS sumber daya seperti bucket Amazon S3 atau tabel DynamoDB, aplikasi memerlukan kredensial keamanan. AWS Namun, mendistribusikan dan menyimpan kredensial keamanan jangka panjang di setiap instans yang Anda luncurkan merupakan tantangan dan potensi risiko keamanan. Alih-alih menggunakan kredensial jangka panjang, seperti kredensial pengguna IAM, sebaiknya Anda membuat peran IAM yang terkait dengan EC2 instans Amazon saat instans diluncurkan. Aplikasi kemudian bisa mendapatkan kredensial keamanan sementara dari instans Amazon EC2 . Anda tidak perlu menyimpan kredensial jangka panjang pada instans. Selain itu, untuk mempermudah pengelolaan kredensial, Anda dapat menentukan hanya satu peran untuk beberapa EC2 instans Amazon; Anda tidak perlu membuat kredensial unik untuk setiap instance.

Untuk potongan templat yang menunjukkan cara meluncurkan instans dengan peran, lihat [Contoh templat IAM role](#).

Note

Aplikasi pada instans yang menggunakan kredensial keamanan sementara dapat memanggil tindakan CloudFormation apa pun. Namun, karena CloudFormation berinteraksi dengan banyak AWS layanan lain, Anda harus memverifikasi bahwa semua layanan yang ingin Anda gunakan mendukung kredensi keamanan sementara. Untuk daftar layanan yang menerima kredensial keamanan sementara, lihat [AWS layanan yang bekerja dengan IAM di Panduan Pengguna IAM](#).

Memberikan akses sementara (akses federasi)

Dalam beberapa kasus, Anda mungkin ingin memberi pengguna tanpa AWS kredensi akses sementara ke Anda. Akun AWS Daripada membuat dan menghapus kredensi jangka panjang kapan pun Anda ingin memberikan akses sementara, gunakan AWS Security Token Service ().AWS STS Misalnya, Anda dapat menggunakan peran IAM. Dari satu peran IAM, Anda dapat secara terprogram membuat dan kemudian mendistribusikan banyak kredensial keamanan sementara (yang mencakup

kunci akses, kunci akses rahasia, dan token keamanan). Kredensi ini memiliki masa pakai yang terbatas, sehingga tidak dapat digunakan untuk mengakses Anda Akun AWS setelah kedaluwarsa. Anda juga dapat membuat beberapa peran IAM untuk memberikan tingkat izin yang berbeda kepada pengguna individu. Peran IAM berguna untuk skenario seperti identitas federasi dan sistem masuk tunggal.

Identitas gabungan adalah identitas berbeda yang dapat Anda gunakan di berbagai sistem. Untuk pengguna perusahaan dengan sistem identitas on-premise yang sudah ditetapkan (seperti LDAP atau Direktori Aktif), Anda dapat menangani semua autentikasi dengan sistem identitas on-premise Anda. Setelah pengguna diautentikasi, Anda memberikan kredensi keamanan sementara dari pengguna atau peran IAM yang sesuai. Misalnya, Anda dapat membuat peran administrator dan peran pengembang, di mana administrator memiliki akses penuh ke AWS akun dan pengembang memiliki izin untuk bekerja hanya dengan tumpukan. CloudFormation Setelah administrator diautentikasi, administrator diberi wewenang untuk mendapatkan kredensial keamanan sementara dari peran administrator. Namun, bagi pengembang, mereka dapat memperoleh kredensial keamanan sementara hanya dari peran pengembang.

Anda juga dapat memberikan akses kepada pengguna federasi ke file. AWS Management Console Setelah pengguna mengautentikasi dengan sistem identitas on-premise, Anda dapat membuat URL sementara secara terprogram yang memberikan akses langsung ke AWS Management Console. Saat pengguna menggunakan URL sementara, mereka tidak perlu masuk AWS karena telah diautentikasi (single sign-on). Selain itu, karena URL dibuat dari kredensial keamanan sementara pengguna, izin yang tersedia dengan kredensial tersebut menentukan izin apa yang dimiliki pengguna di AWS Management Console.

Anda dapat menggunakan beberapa yang berbeda AWS STS APIs untuk menghasilkan kredensial keamanan sementara. Untuk informasi selengkapnya tentang API mana yang akan digunakan, lihat [Membandingkan AWS STS kredensial](#) di Panduan Pengguna IAM.

 Important

Anda tidak dapat bekerja dengan IAM ketika Anda menggunakan kredensial keamanan sementara yang dihasilkan dari API. `GetFederationToken` Sebaliknya, jika Anda perlu bekerja dengan IAM, gunakan kredensial keamanan sementara dari suatu peran.

CloudFormation berinteraksi dengan banyak AWS layanan lainnya. Saat Anda menggunakan kredensial keamanan sementara CloudFormation, verifikasi bahwa semua layanan yang ingin Anda

gunakan mendukung kredensial keamanan sementara. Untuk daftar layanan yang menerima kredensial keamanan sementara, lihat [AWS layanan yang bekerja dengan IAM di Panduan Pengguna IAM](#).

Untuk informasi selengkapnya, lihat sumber daya terkait berikut di Panduan Pengguna IAM:

- [Skenario umum untuk kredensial sementara](#)
- [Aktifkan akses broker identitas khusus ke AWS konsol](#)

Contoh kebijakan berbasis identitas IAM untuk CloudFormation

Secara default, pengguna dan peran tidak memiliki izin untuk membuat atau mengubah sumber daya CloudFormation. Mereka juga tidak dapat melakukan tugas dengan menggunakan AWS Management Console, AWS Command Line Interface (AWS CLI), atau AWS API. Untuk memberikan izin kepada pengguna untuk melakukan tindakan di sumber daya yang mereka perlukan, administrator IAM dapat membuat kebijakan IAM. Administrator kemudian dapat menambahkan kebijakan IAM ke peran, dan pengguna dapat mengambil peran. Untuk informasi selengkapnya, lihat [Mendefinisikan kebijakan berbasis identitas IAM untuk CloudFormation](#).

Contoh berikut menunjukkan pernyataan kebijakan yang dapat Anda gunakan untuk mengizinkan atau menolak izin menggunakan satu atau beberapa CloudFormation tindakan.

Topik

- [Memerlukan URL template tertentu](#)
- [Tolak semua operasi CloudFormation impor](#)
- [Izinkan operasi impor untuk jenis sumber daya tertentu](#)
- [Tolak sumber daya IAM dalam templat tumpukan](#)
- [Izinkan pembuatan tumpukan dengan jenis sumber daya tertentu](#)
- [Kontrol akses berdasarkan tindakan API yang bermutasi sumber daya](#)
- [Batasi operasi set tumpukan berdasarkan Wilayah dan tipe sumber daya](#)
- [Izinkan semua operasi generator IAc](#)

Memerlukan URL template tertentu

Kebijakan berikut memberikan izin untuk hanya menggunakan URL `https://s3.amazonaws.com/amzn-s3-demo-bucket/test.template` templat untuk membuat atau memperbarui tumpukan.

```
{
  "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" ]
        }
      }
    }
  ]
}
```

Tolak semua operasi CloudFormation impor

Kebijakan berikut memberikan izin untuk menyelesaikan semua CloudFormation operasi kecuali operasi impor.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowAllStackOperations",
      "Effect": "Allow",
      "Action": "cloudformation:*",
      "Resource": "*"
    },
    {
      "Sid": "DenyImport",
      "Effect": "Deny",
      "Action": "cloudformation:*",
      "Resource": "*",
      "Condition": {
        "ForAnyValue:StringLike": {
          "cloudformation:ImportResourceTypes": [
            "*"
          ]
        }
      }
    }
  ]
}
```

```

    }
  ]
}
```

Izinkan operasi impor untuk jenis sumber daya tertentu

Kebijakan berikut memberikan izin untuk semua operasi tumpukan, selain operasi impor hanya pada sumber daya tertentu (dalam contoh ini, `AWS::S3::Bucket`)

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowImport",
      "Effect": "Allow",
      "Action": "cloudformation:*",
      "Resource": "*"
      "Condition": {
        "ForAllValues:StringEqualsIgnoreCase": {
          "cloudformation:ImportResourceTypes": [
            "AWS::S3::Bucket"
          ]
        }
      }
    }
  ]
}
```

Tolak sumber daya IAM dalam templat tumpukan

Kebijakan berikut memberikan izin untuk membuat tumpukan tetapi menolak permintaan jika templat tumpukan menyertakan sumber daya apa pun dari layanan IAM. Kebijakan ini juga mengharuskan pengguna untuk menentukan `ResourceTypes` parameter, yang hanya tersedia untuk AWS CLI dan permintaan API. Kebijakan ini menggunakan pernyataan penolakan eksplisit sehingga jika ada kebijakan lain yang memberikan izin tambahan, kebijakan ini selalu tetap berlaku (pernyataan penolakan eksplisit selalu mengesampingkan pernyataan izin eksplisit).

```

{
  "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"
      }
    }
  }
]
}

```

Izinkan pembuatan tumpukan dengan jenis sumber daya tertentu

Kebijakan berikut ini mirip dengan contoh sebelumnya. Kebijakan memberikan izin untuk membuat tumpukan kecuali templat tumpukan menyertakan sumber daya apa pun dari layanan IAM. Hal ini juga mengharuskan pengguna untuk menentukan ResourceTypes parameter, yang hanya tersedia untuk AWS CLI dan permintaan API. Kebijakan ini lebih sederhana, tetapi tidak menggunakan pernyataan penolakan eksplisit. Kebijakan lain, yang memberikan izin tambahan, dapat menimpa kebijakan ini.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect" : "Allow",
      "Action" : [ "cloudformation:CreateStack" ],
      "Resource" : "*",

```



```

    "Condition" : {
      "ForAllValues:StringNotLikeIfExists" : {
        "cloudformation:ResourceTypes" : [ "AWS::IAM::*" ]
      },
      "Null":{
        "cloudformation:ResourceTypes": "false"
      }
    }
  }
]
}

```

Kontrol akses berdasarkan tindakan API yang bermutasi sumber daya

Kebijakan berikut memberikan izin untuk memfilter akses dengan nama tindakan API yang bermutasi sumber daya. Ini digunakan untuk mengontrol pengguna APIs IAM mana yang dapat digunakan untuk menambah atau menghapus tag pada tumpukan atau kumpulan tumpukan. Operasi yang digunakan untuk menambah atau menghapus tag harus ditambahkan sebagai nilai untuk kunci kondisi. Kebijakan berikut memberikan TagResource dan UntagResource izin untuk operasi mutasi. CreateStack

```

{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "CreateActionConditionPolicyForTagUntagResources",
    "Effect": "Allow",
    "Action": [
      "cloudformation:TagResource",
      "cloudformation:UntagResource"
    ],
    "Resource": "*",
    "Condition": {
      "StringEquals": {
        "cloudformation:CreateAction": [
          "CreateStack"
        ]
      }
    }
  ]
}

```

Batasi operasi set tumpukan berdasarkan Wilayah dan tipe sumber daya

Kebijakan berikut memberikan izin set tumpukan yang dikelola layanan. Pengguna dengan kebijakan ini hanya dapat menjalankan operasi pada kumpulan tumpukan dengan templat yang berisi tipe sumber daya Amazon S3 (`AWS::S3::*`) atau jenis `AWS::SES::ConfigurationSet` sumber daya. Jika masuk ke akun manajemen organisasi dengan ID123456789012, pengguna juga hanya dapat melakukan operasi pada kumpulan tumpukan yang menargetkan OU dengan ID `ou-1fsfsrsdsfrewr`, dan hanya dapat melakukan operasi pada kumpulan tumpukan dengan ID `stack-set-id` yang menargetkan ID Akun AWS with `987654321012`.

Operasi set tumpukan gagal jika templat kumpulan tumpukan berisi tipe sumber daya selain yang ditentukan dalam kebijakan, atau jika target penerapan adalah OU atau akun IDs selain yang ditentukan dalam kebijakan untuk akun manajemen dan kumpulan tumpukan terkait.

Pembatasan kebijakan ini hanya berlaku ketika operasi set tumpukan menargetkan `us-east-1`, `us-west-2`, atau `eu-west-2` Wilayah AWS.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudformation:*"
      ],
      "Resource": [
        "arn:aws:cloudformation:*:*:stackset/*",
        "arn:aws:cloudformation:*:*:type/resource/AWS-S3-*",
        "arn:aws:cloudformation:us-west-2:*:type/resource/AWS-SES-ConfigurationSet",
        "arn:aws:cloudformation:*:123456789012:stackset-target/*/ou-1fsfsrsdsfrewr",
        "arn:aws:cloudformation:*:123456789012:stackset-target/stack-set-id/987654321012"
      ],
      "Condition": {
        "ForAllValues:StringEqualsIgnoreCase": {
          "cloudformation:TargetRegion": [
            "us-east-1",
            "us-west-2",
            "eu-west-1"
          ]
        }
      }
    }
  ]
}
```

```

    }
  }
}
]
}

```

Izinkan semua operasi generator IAc

Kebijakan berikut memungkinkan akses ke CloudFormation tindakan yang terkait dengan pemindaian sumber daya generator IAc dan manajemen template. Pernyataan pertama memberikan izin untuk mendeskripsikan, membuat daftar, dan memulai pemindaian sumber daya. Ini juga memungkinkan akses ke izin tambahan yang diperlukan (`cloudformation:GetResource`, `cloudformation:ListResources`, dan `cloudformation:ListTypes`) yang memungkinkan generator IAc untuk mengambil informasi tentang sumber daya dan jenis sumber daya yang tersedia. Pernyataan kedua memberikan izin penuh untuk membuat, menghapus, mendeskripsikan, membuat daftar, dan memperbarui templat yang dihasilkan.

Anda juga harus memberikan izin baca untuk AWS layanan target kepada siapa saja yang akan memindai sumber daya dengan generator IAc. Lihat informasi yang lebih lengkap di [Izin IAM diperlukan untuk memindai sumber daya](#).

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ResourceScanningOperations",
      "Effect": "Allow",
      "Action": [
        "cloudformation:DescribeResourceScan",
        "cloudformation:GetResource",
        "cloudformation:ListResources",
        "cloudformation:ListResourceScanRelatedResources",
        "cloudformation:ListResourceScanResources",
        "cloudformation:ListResourceScans",
        "cloudformation:ListTypes",
        "cloudformation:StartResourceScan"
      ],
      "Resource": "*"
    },
  ],
}

```

```
        "Sid": "TemplateGeneration",
        "Effect": "Allow",
        "Action": [
            "cloudformation:CreateGeneratedTemplate",
            "cloudformation>DeleteGeneratedTemplate",
            "cloudformation:DescribeGeneratedTemplate",
            "cloudformation:GetResource",
            "cloudformation:GetGeneratedTemplate",
            "cloudformation>ListGeneratedTemplates",
            "cloudformation:UpdateGeneratedTemplate"
        ],
        "Resource": "*"
    }
}
```

AWS CloudFormation peran layanan

Peran layanan adalah peran AWS Identity and Access Management (IAM) yang memungkinkan CloudFormation untuk melakukan panggilan ke sumber daya dalam tumpukan atas nama Anda. Anda dapat menentukan peran IAM yang memungkinkan CloudFormation untuk membuat, memperbarui, atau menghapus sumber daya tumpukan Anda. Secara default, CloudFormation gunakan sesi sementara yang dihasilkannya dari kredensi pengguna Anda untuk operasi tumpukan. Jika Anda menentukan peran layanan, CloudFormation gunakan kredensial peran tersebut.

Gunakan peran layanan untuk secara eksplisit menentukan tindakan yang CloudFormation dapat dilakukan, yang mungkin tidak selalu merupakan tindakan yang sama yang dapat Anda atau pengguna lain lakukan. Misalnya, Anda mungkin memiliki hak administratif, tetapi Anda dapat membatasi CloudFormation akses hanya ke EC2 tindakan Amazon.

Anda membuat peran layanan dan kebijakan izinnya dengan layanan IAM. Untuk informasi selengkapnya tentang membuat peran layanan, lihat [Membuat peran untuk mendelegasikan izin ke AWS layanan di Panduan](#) Pengguna IAM. Tentukan CloudFormation (`cloudformation.amazonaws.com`) sebagai layanan yang dapat mengasumsikan peran tersebut.

Untuk mengaitkan peran layanan dengan tumpukan, tentukan peran saat Anda membuat tumpukan. Untuk detailnya, lihat [Konfigurasi opsi tumpukan](#). Anda juga dapat mengubah peran layanan saat memperbarui tumpukan di konsol, atau [DeleteStack](#) tumpukan melalui API. Sebelum Anda menentukan peran layanan, pastikan bahwa Anda memiliki izin untuk memberikannya (`iam:PassRole`). Izin `iam:PassRole` menentukan peran yang dapat Anda gunakan. Untuk

informasi selengkapnya, lihat [Memberikan izin pengguna untuk meneruskan peran ke AWS layanan](#) di Panduan Pengguna IAM.

Important

Saat Anda menentukan peran layanan, CloudFormation selalu gunakan peran itu untuk semua operasi yang dilakukan pada tumpukan tersebut. Penghapusan peran layanan yang melekat pada tumpukan setelah tumpukan dibuat tidak dapat dilakukan. Pengguna lain yang memiliki izin untuk melakukan operasi pada tumpukan ini dapat menggunakan peran ini, terlepas dari apakah pengguna tersebut memiliki `iam:PassRole` izin atau tidak. Jika peran tersebut mencakup izin yang seharusnya tidak dimiliki pengguna, Anda dapat secara tidak sengaja meningkatkan izin pengguna. Pastikan peran memberikan keistimewaan paling rendah. Untuk informasi selengkapnya, lihat [Menerapkan izin hak istimewa paling sedikit di Panduan Pengguna](#) IAM.

Pencegahan "confused deputy" lintas layanan

Masalah "confused deputy" adalah masalah keamanan saat entitas yang tidak memiliki izin untuk melakukan suatu tindakan dapat memaksa entitas yang memiliki hak akses lebih tinggi untuk melakukan tindakan tersebut. Pada tahun AWS, peniruan lintas layanan dapat mengakibatkan masalah wakil yang membingungkan. Peniruan identitas lintas layanan dapat terjadi ketika satu layanan (layanan yang dipanggil) memanggil layanan lain (layanan yang dipanggil). Layanan panggilan dapat dimanipulasi untuk menggunakan izinnya untuk bertindak atas sumber daya pelanggan lain dengan cara yang seharusnya tidak memiliki izin untuk mengakses. Untuk mencegah hal ini, AWS menyediakan alat yang membantu Anda melindungi data Anda untuk semua layanan dengan prinsip layanan yang telah diberikan akses ke sumber daya di akun Anda.

Kami merekomendasikan menggunakan [aws:SourceArn](#) dan [aws:SourceAccount](#) kunci konteks kondisi global dalam kebijakan sumber daya untuk membatasi izin yang AWS CloudFormation memberikan layanan lain ke sumber daya tertentu, seperti CloudFormation ekstensi. Gunakan `aws:SourceArn` jika Anda ingin hanya satu sumber daya yang akan dikaitkan dengan akses lintas layanan. Gunakan `aws:SourceAccount` jika Anda ingin mengizinkan sumber daya apa pun di akun tersebut dikaitkan dengan penggunaan lintas layanan.

Pastikan bahwa nilai `aws:SourceArn` adalah ARN dari sumber daya yang CloudFormation menyimpan.

Cara paling efektif untuk melindungi dari masalah "confused deputy" adalah dengan menggunakan kunci konteks kondisi global `aws:SourceArn` dengan ARN lengkap sumber daya. Jika Anda tidak mengetahui ARN lengkap sumber daya atau jika Anda menentukan beberapa sumber daya, gunakan kunci kondisi konteks `aws:SourceArn` global dengan wildcard (*) untuk bagian ARN yang tidak diketahui. Misalnya, `arn:aws:cloudformation:*:123456789012:*`.

Jika `aws:SourceArn` nilai tidak berisi ID akun, Anda harus menggunakan kedua kunci konteks kondisi global untuk membatasi izin.

Contoh berikut menunjukkan bagaimana Anda dapat menggunakan kunci konteks kondisi `aws:SourceAccount` global `aws:SourceArn` dan CloudFormation untuk mencegah masalah wakil yang membingungkan.

Contoh kebijakan kepercayaan yang menggunakan **`aws:SourceArn`** dan kunci **`aws:SourceAccount`** kondisi

Untuk layanan registri CloudFormation, lakukan panggilan ke AWS Security Token Service (AWS STS) untuk mengambil peran layanan di akun Anda. Peran ini dikonfigurasi untuk `ExecutionRoleArn` di [RegisterType](#) operasi dan `LogRoleArn` diatur dalam [LoggingConfig](#) operasi. Untuk informasi selengkapnya, lihat [Mengonfigurasi peran eksekusi dengan izin IAM dan kebijakan kepercayaan untuk akses ekstensi publik](#).

Contoh kebijakan kepercayaan peran ini menggunakan pernyataan kondisi untuk membatasi `AssumeRole` kemampuan peran layanan hanya pada tindakan pada CloudFormation ekstensi tertentu di akun yang ditentukan. `aws:SourceAccount` Kondisi `aws:SourceArn` dan dievaluasi secara independen. Setiap permintaan untuk menggunakan peran layanan harus memenuhi kedua kondisi tersebut.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "resources.cloudformation.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
```

```
    "aws:SourceAccount": "123456789012"
  },
  "StringLike": {
    "aws:SourceArn": "arn:aws:cloudformation:us-east-1:123456789012:type/
resource/Organization-Service-Resource/*"
  }
}
]
```

Informasi tambahan

Misalnya kebijakan yang menggunakan kunci konteks kondisi `aws:SourceAccount` global `aws:SourceArn` dan untuk peran layanan yang digunakan oleh StackSets, lihat [Siapkan kunci global untuk mengurangi masalah wakil yang membingungkan](#).

Untuk informasi selengkapnya, lihat [Memperbarui kebijakan kepercayaan peran](#) di Panduan Pengguna IAM.

Permintaan sesi akses teruskan (FAS) dan evaluasi izin

Saat membuat, memperbarui, dan menghapus CloudFormation tumpukan, pengguna dapat secara opsional menentukan peran IAM ARN. Jika tidak ada peran yang disediakan, CloudFormation gunakan mekanisme layanan defaultnya untuk berinteraksi dengan AWS layanan lain. Dalam skenario ini, penelepon harus memiliki izin yang diperlukan untuk sumber daya yang dikelola. Atau, ketika pengguna memasok peran IAM mereka sendiri, CloudFormation akan mengambil peran itu untuk melakukan interaksi layanan atas nama mereka.

Terlepas dari apakah pengguna menyediakan peran IAM, CloudFormation menghasilkan token FAS cakupan bawah baru untuk setiap operasi sumber daya. Akibatnya, [kunci kondisi terkait FAS](#), termasuk `aws:ViaAWSService`, diisi di kedua skenario.

Penggunaan FAS mempengaruhi bagaimana kebijakan IAM dievaluasi selama operasi.

CloudFormation Saat membuat tumpukan dengan templat yang menyertakan sumber daya yang dipengaruhi oleh kunci kondisi terkait FAS, penolakan izin dapat terjadi.

Contoh kebijakan IAM

Pertimbangkan kebijakan IAM berikut. `Statement2` akan secara konsisten mencegah penciptaan sumber `AWS::KMS::Key` di CloudFormation. Pembatasan akan diberlakukan secara konsisten,

terlepas dari apakah peran IAM disediakan atau tidak selama operasi tumpukan. Ini karena kunci `aws:ViaAWSService` kondisi selalu disetel `true` karena penggunaan FAS.

```
{
  "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"
        }
      }
    }
  ]
}
```

Contoh template tumpukan

Misalnya, ketika pengguna membuat tumpukan dengan contoh template berikut, `aws:ViaAWSService` disetel ke `true`, dan izin peran akan diganti oleh kebijakan FAS. Pembuatan tumpukan akan dipengaruhi oleh `Statement2` kebijakan IAM, yang menyangkal tindakan tersebut `CreateKey`. Ini menghasilkan kesalahan izin ditolak.

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: '*'
```

Untuk informasi selengkapnya tentang FAS, lihat [Teruskan sesi akses](#) di Panduan Pengguna IAM.

Note

Sebagian besar sumber daya mematuhi perilaku ini. Namun, jika Anda mengalami keberhasilan atau kegagalan yang tidak terduga saat membuat, memperbarui, atau menghapus sumber daya, dan kebijakan IAM Anda menyertakan kunci kondisi terkait FAS, kemungkinan sumber daya tersebut milik sebagian kecil sumber daya yang tidak mengikuti pola standar ini.

Logging panggilan AWS CloudFormation API dengan AWS CloudTrail

AWS CloudFormation terintegrasi dengan AWS CloudTrail, layanan yang menyediakan catatan tindakan yang diambil oleh pengguna, peran, atau AWS layanan di CloudFormation. CloudTrail menangkap semua panggilan API untuk CloudFormation sebagai peristiwa, termasuk panggilan dari CloudFormation konsol dan dari panggilan kode ke CloudFormation APIs. Jika Anda membuat jejak, Anda dapat mengaktifkan pengiriman CloudTrail acara secara berkelanjutan ke bucket Amazon S3, termasuk acara untuk CloudFormation. Jika Anda tidak mengonfigurasi jejak, Anda masih dapat

melihat peristiwa terbaru di CloudTrail konsol dalam Riwayat acara. Dengan menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan permintaan yang dibuat CloudFormation, alamat IP dari mana permintaan dibuat, siapa yang membuat permintaan, kapan dibuat, dan detail tambahan.

Untuk mempelajari selengkapnya CloudTrail, lihat [Panduan AWS CloudTrail Pengguna](#).

Topik

- [CloudFormation informasi di CloudTrail](#)
- [Memahami entri file CloudFormation log](#)

CloudFormation informasi di CloudTrail

CloudTrail diaktifkan di AWS akun Anda saat Anda membuat akun. Ketika aktivitas terjadi di CloudFormation, aktivitas tersebut dicatat dalam suatu CloudTrail peristiwa bersama dengan peristiwa AWS layanan lainnya dalam riwayat Acara. Anda dapat melihat, mencari, dan mengunduh peristiwa terbaru di akun AWS. Untuk informasi selengkapnya, lihat [Melihat peristiwa dengan riwayat CloudTrail acara](#).

Untuk catatan peristiwa yang sedang berlangsung di AWS akun Anda, termasuk acara untuk CloudFormation, buat jejak. Jejak memungkinkan CloudTrail untuk mengirimkan file log ke bucket Amazon S3. Secara default, ketika Anda membuat jejak di konsol, jejak diterapkan ke semua Region. Jejak mencatat peristiwa dari semua Wilayah di AWS partisi dan mengirimkan file log ke bucket Amazon S3 yang Anda tentukan. Selain itu, Anda dapat mengonfigurasi AWS layanan lain untuk menganalisis lebih lanjut dan menindaklanjuti data peristiwa yang dikumpulkan dalam CloudTrail log. Untuk informasi selengkapnya, lihat:

- [Ikhtisar untuk membuat jejak](#)
- [CloudTrail layanan dan integrasi yang didukung](#)
- [Mengonfigurasi notifikasi Amazon SNS untuk CloudTrail](#)
- [Menerima file CloudTrail log dari beberapa Wilayah](#) dan [Menerima file CloudTrail log dari beberapa akun](#)

Semua CloudFormation tindakan dicatat oleh CloudTrail dan didokumentasikan dalam [Referensi AWS CloudFormation API](#). Misalnya, panggilan ke `CreateStackDeleteStack`, dan `ListStacks` bagian menghasilkan entri dalam file CloudTrail log.

Setiap entri peristiwa atau log berisi informasi tentang siapa yang membuat permintaan tersebut. Informasi identitas membantu Anda menentukan hal berikut ini:

- Apakah permintaan tersebut dibuat dengan kredensial root atau pengguna IAM.
- Apakah permintaan tersebut dibuat dengan kredensial keamanan sementara untuk satu peran atau pengguna terfederasi.
- Apakah permintaan itu dibuat oleh AWS layanan lain.

Untuk informasi selengkapnya, lihat [Elemen userIdentity CloudTrail](#).

Memahami entri file CloudFormation log

Trail adalah konfigurasi yang memungkinkan pengiriman peristiwa sebagai file log ke bucket Amazon S3 yang Anda tentukan. CloudTrail file log berisi satu atau lebih entri log. Peristiwa mewakili permintaan tunggal dari sumber mana pun dan mencakup informasi tentang operasi yang diminta, tanggal dan waktu operasi, parameter permintaan, dan sebagainya. CloudTrail file log bukanlah jejak tumpukan yang diurutkan dari panggilan API publik, jadi file tersebut tidak muncul dalam urutan tertentu.

Contoh berikut menunjukkan entri CloudTrail log yang menunjukkan CreateStack operasi. Operasi ini dilakukan oleh pengguna IAM bernama Alice.

Note

Hanya nama kunci parameter input yang dicatat. Nilai parameter tidak dicatat.

```
{
  "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"
}

```

Contoh berikut menunjukkan bahwa Alice memanggil UpdateStack operasi pada my-test-stack tumpukan:

```

{
  "eventVersion": "1.01",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDAABCDEFGHijklmnopq",
    "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"
}

```

Contoh berikut menunjukkan bahwa Alice menyebut ListStacks operasi itu.

```

{
  "eventVersion": "1.01",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDAABCDEFGHijklmOPQ",
    "arn": "arn:aws:iam::012345678910:user/Alice",
    "accountId": "012345678910",
    "accessKeyId": "AKIDEXAMPLE",
    "userName": "Alice"
  },
  "eventTime": "2014-03-24T21:03:16Z",
  "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"
}
```

Contoh berikut menunjukkan bahwa Alice memanggil DescribeStacks operasi pada my-test-stack tumpukan.

```
{
  "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"
}
```

Contoh berikut menunjukkan bahwa Alice memanggil DeleteStack operasi pada my-test-stack tumpukan.

```
{
  "eventVersion": "1.01",
  "userIdentity": {
    "type": "IAMUser",
```

```
{
  "principalId": "AIDAABCDEFGHijklmnopq",
  "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"
}
```

Keamanan infrastruktur di AWS CloudFormation

Sebagai layanan terkelola, AWS CloudFormation dilindungi oleh keamanan jaringan AWS global. Untuk informasi tentang layanan AWS keamanan dan cara AWS melindungi infrastruktur, lihat [Keamanan AWS Cloud](#). Untuk mendesain AWS lingkungan Anda menggunakan praktik terbaik untuk keamanan infrastruktur, lihat [Perlindungan Infrastruktur dalam Kerangka Kerja](#) yang AWS Diarsiteksikan dengan Baik Pilar Keamanan.

Anda menggunakan panggilan API yang AWS dipublikasikan untuk mengakses AWS CloudFormation melalui jaringan. Klien harus mendukung hal-hal berikut:

- Keamanan Lapisan Pengangkutan (TLS). Kami mensyaratkan TLS 1.2 dan menganjurkan TLS 1.3.
- Sandi cocok dengan sistem kerahasiaan maju sempurna (perfect forward secrecy, PFS) seperti DHE (Ephemeral Diffie-Hellman) atau ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Sebagian besar sistem modern seperti Java 7 dan versi lebih baru mendukung mode-mode ini.

Selain itu, permintaan harus ditandatangani menggunakan ID kunci akses dan kunci akses rahasia yang terkait dengan prinsipal IAM. Atau Anda dapat menggunakan [AWS Security Token Service](#) (AWS STS) untuk menghasilkan kredensial keamanan sementara untuk menandatangani permintaan.

Anda dapat memanggil operasi API ini dari lokasi jaringan mana pun, AWS CloudFormation tetapi mendukung kebijakan akses berbasis sumber daya, yang dapat mencakup pembatasan berdasarkan alamat IP sumber. Anda juga dapat menggunakan AWS CloudFormation kebijakan untuk mengontrol akses dari titik akhir Amazon Virtual Private Cloud (Amazon VPC) tertentu atau spesifik. VPCs Secara efektif, ini mengisolasi akses jaringan ke AWS CloudFormation sumber daya tertentu hanya dari VPC tertentu dalam AWS jaringan.

Ketahanan di AWS CloudFormation

Infrastruktur AWS global dibangun di sekitar AWS Wilayah dan Zona Ketersediaan. AWS Wilayah menyediakan beberapa Availability Zone yang terpisah secara fisik dan terisolasi, yang terhubung dengan latensi rendah, throughput tinggi, dan jaringan yang sangat redundan. Dengan Zona Ketersediaan, Anda dapat merancang serta mengoperasikan aplikasi dan basis data yang secara otomatis melakukan fail over di antara zona tanpa gangguan. Zona Ketersediaan memiliki ketersediaan dan toleransi kesalahan yang lebih baik, dan dapat diskalakan dibandingkan infrastruktur pusat data tunggal atau multi tradisional.

Untuk informasi selengkapnya tentang AWS Wilayah dan Availability Zone, lihat [Infrastruktur AWS Global](#).

Validasi kepatuhan untuk AWS CloudFormation

Untuk mempelajari apakah an Layanan AWS berada dalam lingkup program kepatuhan tertentu, lihat [Layanan AWS di Lingkup oleh Program Kepatuhan Layanan AWS](#) dan pilih program kepatuhan yang Anda minati. Untuk informasi umum, lihat [Program AWS Kepatuhan Program AWS](#) .

Anda dapat mengunduh laporan audit pihak ketiga menggunakan AWS Artifact. Untuk informasi selengkapnya, lihat [Mengunduh Laporan di AWS Artifact](#) .

Tanggung jawab kepatuhan Anda saat menggunakan Layanan AWS ditentukan oleh sensitivitas data Anda, tujuan kepatuhan perusahaan Anda, dan hukum dan peraturan yang berlaku. AWS menyediakan sumber daya berikut untuk membantu kepatuhan:

- [Kepatuhan dan Tata Kelola Keamanan](#) – Panduan implementasi solusi ini membahas pertimbangan arsitektur serta memberikan langkah-langkah untuk menerapkan fitur keamanan dan kepatuhan.
- [Referensi Layanan yang Memenuhi Syarat HIPAA](#) — Daftar layanan yang memenuhi syarat HIPAA. Tidak semua memenuhi Layanan AWS syarat HIPAA.

- [AWS Sumber Daya AWS](#) — Kumpulan buku kerja dan panduan ini mungkin berlaku untuk industri dan lokasi Anda.
- [AWS Panduan Kepatuhan Pelanggan](#) - Memahami model tanggung jawab bersama melalui lensa kepatuhan. Panduan ini merangkum praktik terbaik untuk mengamankan Layanan AWS dan memetakan panduan untuk kontrol keamanan di berbagai kerangka kerja (termasuk Institut Standar dan Teknologi Nasional (NIST), Dewan Standar Keamanan Industri Kartu Pembayaran (PCI), dan Organisasi Internasional untuk Standardisasi (ISO)).
- [Mengevaluasi Sumber Daya dengan Aturan](#) dalam Panduan AWS Config Pengembang — AWS Config Layanan menilai seberapa baik konfigurasi sumber daya Anda mematuhi praktik internal, pedoman industri, dan peraturan.
- [AWS Security Hub](#)— Ini Layanan AWS memberikan pandangan komprehensif tentang keadaan keamanan Anda di dalamnya AWS. Security Hub menggunakan kontrol keamanan untuk sumber daya AWS Anda serta untuk memeriksa kepatuhan Anda terhadap standar industri keamanan dan praktik terbaik. Untuk daftar layanan dan kontrol yang didukung, lihat [Referensi kontrol Security Hub](#).
- [Amazon GuardDuty](#) — Ini Layanan AWS mendeteksi potensi ancaman terhadap beban kerja Akun AWS, kontainer, dan data Anda dengan memantau lingkungan Anda untuk aktivitas mencurigakan dan berbahaya. GuardDuty dapat membantu Anda mengatasi berbagai persyaratan kepatuhan, seperti PCI DSS, dengan memenuhi persyaratan deteksi intrusi yang diamanatkan oleh kerangka kerja kepatuhan tertentu.
- [AWS Audit Manager](#) Ini Layanan AWS membantu Anda terus mengaudit AWS penggunaan Anda untuk menyederhanakan cara Anda mengelola risiko dan kepatuhan terhadap peraturan dan standar industri.

Konfigurasi dan analisis kerentanan di AWS CloudFormation

Konfigurasi dan kontrol TI adalah tanggung jawab bersama antara AWS dan Anda, pelanggan kami. Untuk informasi selengkapnya, lihat [model tanggung jawab AWS bersama](#).

Praktik terbaik keamanan untuk CloudFormation

AWS CloudFormation menyediakan sejumlah fitur keamanan untuk dipertimbangkan saat Anda mengembangkan dan menerapkan kebijakan keamanan Anda sendiri. Praktik terbaik berikut adalah pedoman umum dan tidak mewakili solusi keamanan yang lengkap. Karena praktik terbaik

ini mungkin tidak sesuai atau tidak memadai untuk lingkungan Anda, perlakukan itu sebagai pertimbangan yang bermanfaat, bukan sebagai resep.

Topik

- [Gunakan IAM untuk mengontrol akses](#)
- [Jangan menanamkan kredensial dalam templat Anda](#)
- [Gunakan AWS CloudTrail untuk mencatat CloudFormation panggilan](#)

Gunakan IAM untuk mengontrol akses

IAM adalah AWS layanan yang dapat Anda gunakan untuk mengelola pengguna dan izin mereka. AWS Anda dapat menggunakan IAM CloudFormation untuk menentukan CloudFormation tindakan apa yang dapat dilakukan pengguna, seperti melihat templat tumpukan, membuat tumpukan, atau menghapus tumpukan. Selain itu, siapa pun yang mengelola CloudFormation tumpukan akan memerlukan izin untuk sumber daya di dalam tumpukan tersebut. Misalnya, jika pengguna ingin menggunakan CloudFormation untuk meluncurkan, memperbarui, atau menghentikan EC2 instans Amazon, mereka harus memiliki izin untuk memanggil tindakan Amazon EC2 yang relevan.

Dalam kebanyakan kasus, pengguna memerlukan akses penuh untuk mengelola semua sumber daya dalam templat. CloudFormation membuat panggilan untuk membuat, memodifikasi, dan menghapus sumber daya atas nama mereka. Untuk memisahkan izin antara pengguna dan CloudFormation layanan, gunakan peran layanan. CloudFormation menggunakan kebijakan peran layanan untuk melakukan panggilan, bukan kebijakan pengguna. Untuk informasi selengkapnya, lihat [AWS CloudFormation peran layanan](#).

Jangan menanamkan kredensial dalam templat Anda

Daripada menyematkan informasi sensitif di CloudFormation template Anda, kami sarankan Anda menggunakan referensi dinamis di template tumpukan Anda.

Referensi dinamis menyediakan cara yang ringkas dan ampuh bagi Anda untuk mereferensikan nilai eksternal yang disimpan dan dikelola di layanan lain, seperti AWS Systems Manager Parameter Store atau AWS Secrets Manager. Ketika Anda menggunakan referensi dinamis, CloudFormation mengambil nilai referensi yang ditentukan saat diperlukan selama operasi tumpukan dan set perubahan, dan meneruskan nilai ke sumber daya yang sesuai. Namun, CloudFormation tidak pernah menyimpan nilai referensi yang sebenarnya. Untuk informasi selengkapnya, lihat [Dapatkan nilai yang disimpan di layanan lain menggunakan referensi dinamis](#).

[AWS Secrets Manager](#) membantu Anda mengenkripsi, menyimpan, dan mengambil kredensial untuk basis data dan layanan lainnya dengan aman. [AWS Systems Manager Parameter Store](#) menyediakan penyimpanan hierarkis yang aman untuk pengelolaan data konfigurasi.

Untuk informasi lebih lanjut tentang menentukan parameter templat, lihat [CloudFormation Parameterssintaks template](#).

Gunakan AWS CloudTrail untuk mencatat CloudFormation panggilan

AWS CloudTrail melacak siapa pun yang melakukan panggilan CloudFormation API di Akun AWS. Panggilan API dicatat setiap kali ada yang menggunakan CloudFormation API, CloudFormation konsol, konsol back-end, atau CloudFormation AWS CLI perintah. Aktifkan logging dan tentukan bucket Amazon S3 untuk menyimpan log. Dengan begitu, jika Anda perlu melakukannya, Anda dapat meng-audit siapa yang membuat panggilan CloudFormation di akun Anda. Untuk informasi selengkapnya, lihat [Logging panggilan AWS CloudFormation API dengan AWS CloudTrail](#).

Akses CloudFormation menggunakan endpoint antarmuka ()AWS PrivateLink

Anda dapat menggunakan AWS PrivateLink untuk membuat koneksi pribadi antara VPC Anda dan CloudFormation. Anda dapat mengakses CloudFormation seolah-olah itu ada di VPC Anda, tanpa menggunakan gateway internet, perangkat NAT, koneksi VPN, atau koneksi. AWS Direct Connect Instans di VPC Anda tidak memerlukan alamat IP publik untuk mengakses CloudFormation.

Anda membuat koneksi pribadi ini dengan membuat titik akhir antarmuka, yang didukung oleh AWS PrivateLink. Kami membuat antarmuka jaringan endpoint di setiap subnet yang Anda aktifkan untuk titik akhir antarmuka. Ini adalah antarmuka jaringan yang dikelola pemohon yang berfungsi sebagai titik masuk untuk lalu lintas yang ditakdirkan CloudFormation.

CloudFormation mendukung panggilan ke semua tindakan API-nya melalui titik akhir antarmuka.

Pertimbangan untuk titik akhir CloudFormation VPC

Sebelum Anda menyiapkan titik akhir antarmuka, pertama-tama pastikan Anda telah memenuhi prasyarat dalam [Akses layanan AWS menggunakan topik titik akhir VPC antarmuka](#) di Panduan.AWS PrivateLink.

Prasyarat dan pertimbangan tambahan berikut berlaku saat menyiapkan titik akhir antarmuka untuk: CloudFormation

- Jika Anda memiliki sumber daya dalam VPC yang harus menanggapi permintaan sumber daya khusus atau kondisi tunggu, pastikan mereka memiliki akses ke bucket Amazon CloudFormation S3 khusus yang diperlukan. CloudFormation memiliki bucket S3 di setiap Wilayah untuk memantau respons terhadap permintaan [sumber daya khusus](#) atau kondisi [tunggu](#). Jika templat menyertakan sumber daya kustom atau kondisi menunggu di VPC, kebijakan akhir VPC endpoint harus mengizinkan pengguna untuk mengirim respons ke bucket berikut:
 - Untuk sumber daya kustom, izinkan lalu lintas ke bucket `cloudformation-custom-resource-response-region`. Saat menggunakan sumber daya khusus, Wilayah AWS nama tidak mengandung tanda hubung. Misalnya, `uswest2`.
 - Untuk kondisi menunggu, izinkan lalu lintas ke bucket `cloudformation-waitcondition-region`. Saat menggunakan kondisi tunggu, Wilayah AWS nama memang mengandung tanda hubung. Misalnya, `us-west-2`.

Jika kebijakan endpoint memblokir lalu lintas ke bucket ini, tidak CloudFormation akan menerima tanggapan dan operasi stack gagal. Misalnya, jika Anda memiliki sumber daya VPC di `us-west-2` Region yang harus menanggapi kondisi menunggu, sumber daya harus dapat mengirim respons ke bucket `cloudformation-waitcondition-us-west-2`.

Untuk daftar Wilayah AWS tempat yang CloudFormation tersedia saat ini, lihat halaman [AWS CloudFormation titik akhir dan kuota](#) di halaman. Referensi Umum Amazon Web Services

- Titik akhir VPC saat ini tidak mendukung permintaan Lintas wilayah — pastikan Anda membuat titik akhir di Wilayah yang sama tempat Anda berencana untuk mengeluarkan panggilan API. CloudFormation
- Titik akhir VPC hanya mendukung DNS yang disediakan Amazon melalui Route 53. Jika Anda ingin menggunakan DNS Anda sendiri, Anda dapat menggunakan penerusan DNS bersyarat. Untuk informasi selengkapnya, lihat [set opsi DHCP di Amazon VPC](#) di Panduan Pengguna Amazon VPC.
- Grup keamanan yang dilampirkan ke VPC endpoint harus mengizinkan koneksi masuk pada port 443 dari subnet privat VPC.

Buat VPC endpoint antarmuka untuk CloudFormation

Anda dapat membuat titik akhir VPC untuk CloudFormation menggunakan konsol VPC Amazon atau (). AWS Command Line Interface AWS CLI Untuk informasi selengkapnya, lihat [Membuat titik akhir VPC](#) di Panduan Pengguna AWS PrivateLink .

Buat titik akhir antarmuka untuk CloudFormation menggunakan nama layanan berikut:

- `com.amazonaws. region.cloudformasi`

Jika Anda mengaktifkan DNS pribadi untuk titik akhir antarmuka, Anda dapat membuat permintaan API untuk CloudFormation menggunakan nama DNS Regional default. Misalnya, `cloudformation.us-east-1.amazonaws.com`.

Membuat kebijakan VPC endpoint untuk CloudFormation

Kebijakan endpoint adalah sumber daya IAM yang dapat Anda lampirkan ke titik akhir antarmuka. Kebijakan endpoint default memungkinkan akses penuh CloudFormation melalui titik akhir antarmuka. Untuk mengontrol akses yang diizinkan CloudFormation dari VPC Anda, lampirkan kebijakan titik akhir kustom ke titik akhir antarmuka.

kebijakan titik akhir mencantumkan informasi berikut:

- Prinsipal yang dapat melakukan tindakan (Akun AWS, pengguna IAM, dan peran IAM).
- Tindakan yang dapat dilakukan.
- Sumber daya untuk melakukan tindakan.

Untuk informasi selengkapnya, lihat [Mengontrol akses ke titik akhir VPC menggunakan kebijakan titik akhir](#) di Panduan.AWS PrivateLink

Contoh: Kebijakan VPC endpoint untuk tindakan CloudFormation

Berikut ini adalah contoh kebijakan endpoint untuk CloudFormation. Saat dilampirkan ke titik akhir, kebijakan ini memberikan akses ke CloudFormation tindakan yang tercantum untuk semua prinsipal di semua sumber daya. Contoh berikut menolak semua pengguna izin untuk membuat tumpukan melalui titik akhir VPC, dan memungkinkan akses penuh ke semua tindakan lain pada layanan. CloudFormation

```
{
```

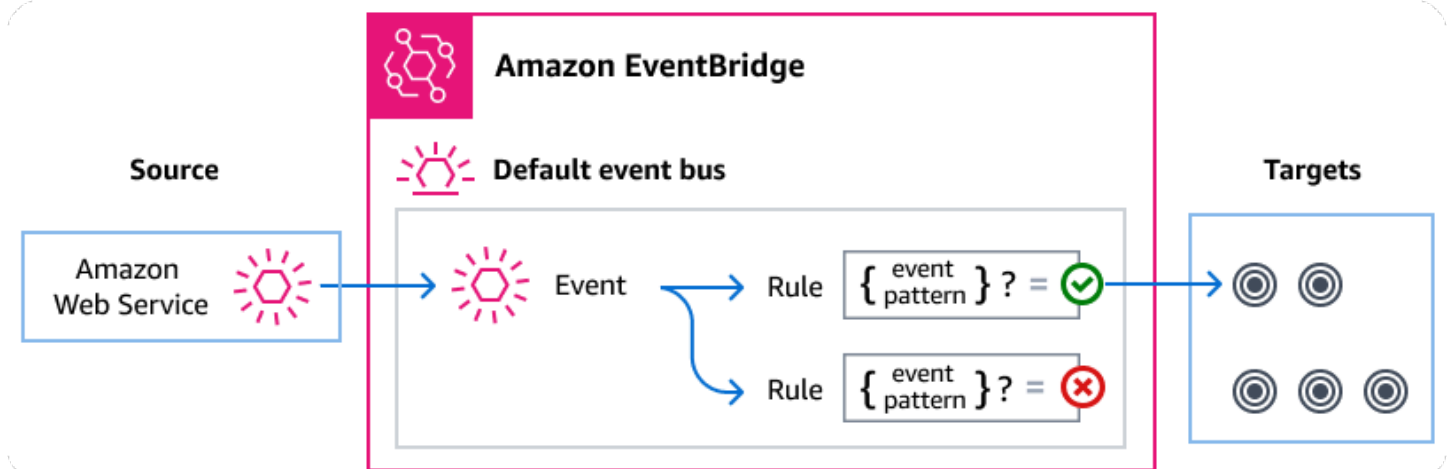
```
"Statement": [  
  {  
    "Action": "cloudformation:*",  
    "Effect": "Allow",  
    "Principal": "*",  
    "Resource": "*"   
  },  
  {  
    "Action": "cloudformation:CreateStack",  
    "Effect": "Deny",  
    "Principal": "*",  
    "Resource": "*"   
  }  
]  
}
```

Memantau CloudFormation dan menyinkronkan peristiwa Git dengan EventBridge

Amazon EventBridge adalah layanan tanpa server yang menggunakan peristiwa untuk menghubungkan komponen aplikasi bersama-sama, sehingga memudahkan Anda untuk membangun aplikasi berbasis peristiwa yang dapat diskalakan. Arsitektur berbasis peristiwa adalah gaya membangun sistem perangkat lunak yang digabungkan secara longgar yang bekerja sama dengan memancarkan dan menanggapi peristiwa. Peristiwa mewakili perubahan dalam sumber daya atau lingkungan.

Seperti banyak AWS layanan, CloudFormation menghasilkan dan mengirim acara ke bus acara EventBridge default, yang secara otomatis disediakan di setiap Akun AWS. Bus acara adalah router yang menerima acara dan mengirimkannya ke nol atau lebih tujuan, atau target. Aturan yang Anda tentukan untuk bus acara mengevaluasi peristiwa saat mereka tiba. Setiap aturan memeriksa apakah suatu peristiwa cocok dengan pola acara aturan. Jika acara tidak cocok, bus acara mengirimkan acara ke target yang ditentukan.

Untuk informasi selengkapnya, lihat [Memulai Amazon EventBridge](#) di Panduan EventBridge Pengguna Amazon.



Topik

- [CloudFormation dan ikhtisar acara sinkronisasi Git](#)
- [EventBridge Izin Amazon](#)
- [Membuat pola acara khusus untuk EventBridge aturan](#)
- [CloudFormation referensi detail acara](#)

CloudFormation dan ikhtisar acara sinkronisasi Git

CloudFormation mengirimkan peristiwa ke EventBridge setiap kali operasi membuat, memperbarui, menghapus, atau deteksi drift dilakukan pada tumpukan. CloudFormation juga mengirimkan peristiwa ke EventBridge perubahan status ke set tumpukan dan instance set tumpukan. Anda dapat menggunakan EventBridge aturan untuk merutekan peristiwa ke target yang ditentukan. Acara-acara ini dijamin akan dikirimkan, dan mereka mungkin dikirim rusak.

Karena CloudFormation peristiwa mewakili perubahan pada tumpukan atau kumpulan tumpukan dan sumber dayanya, Anda dapat menggunakannya untuk memulai alur kerja yang terkait dengan peristiwa masing-masing. Misalnya:

- Buat tumpukan atau tumpukan set tag tertentu pada semua sumber daya yang disediakan. CloudFormation
- Buat asosiasi antara kumpulan CloudFormation tumpukan atau tumpukan dan Manajer WorkSpaces Aplikasi Amazon (Amazon WAM).
- Tentukan asosiasi dengan AppRegistry kumpulan tumpukan atau tumpukan yang dibuat.

Peristiwa berikut dihasilkan oleh CloudFormation dan dikirim ke bus acara default di EventBridge. Untuk informasi selengkapnya, lihat [???](#).

Jenis peristiwa	Deskripsi	Sumber peristiwa
Perubahan Status Sumber Daya	Setiap pembaruan dilakukan pada tumpukan yang mengubah properti sumber daya yang mendasarinya. Untuk daftar lengkap jenis AWS sumber daya yang didukung, lihat referensi jenis AWS sumber daya dan properti.	AWS CloudFormation
Perubahan Status Stack	Merupakan perubahan status ke tumpukan tertentu. Untuk detail kode, lihat Kode status tumpukan.	AWS CloudFormation
Perubahan Status Deteksi Drift	Merupakan pembaruan deteksi drift yang dimulai pengguna pada tumpukan tertentu.	AWS CloudFormation

Jenis peristiwa	Deskripsi	Sumber peristiwa
	Untuk daftar lengkap tipe yang sepenuhnya dapat berubah dan tidak dapat diubah yang mendukung deteksi drift, lihat Dukungan jenis sumber daya	
StackSetPerubahan Status	Merupakan perubahan status ke set tumpukan yang diberikan.	AWS CloudFormation
StackSet Perubahan Status Instans Stack	Merupakan perubahan status ke instance StackSet stack tertentu. Untuk detail kode, lihat Kode status instance tumpukan .	AWS CloudFormation
StackSet status operasi	Merupakan perubahan status ke StackSet operasi tertentu. Untuk detail kode, lihat StackSets kode status .	AWS CloudFormation

Selain itu, sinkronisasi AWS CloudFormation Git mengirimkan peristiwa untuk perubahan status untuk sinkronisasi repositori dan sinkronisasi sumber daya ke EventBridge

Peristiwa sinkronisasi Git berikut dihasilkan oleh CodeConnections dan dikirim ke bus acara default di EventBridge. Untuk informasi selengkapnya, lihat [???](#).

Jenis peristiwa	Deskripsi	Sumber peristiwa
Perubahan status sinkronisasi repositori	Merupakan perubahan status ke sinkronisasi repositori Git.	AWS CodeConnections
Perubahan status sinkronisasi sumber daya	Merupakan perubahan status ke sinkronisasi sumber daya Git.	AWS CodeConnections

EventBridge Izin Amazon

CloudFormation tidak memerlukan izin tambahan untuk mengirimkan acara ke EventBridge. Peristiwa berisi informasi yang sudah tersedia melalui CloudFormation operasi API.

Target yang Anda tentukan mungkin memerlukan izin atau konfigurasi tertentu. Untuk detail selengkapnya tentang penggunaan layanan khusus untuk target, lihat [EventBridge Target](#) Amazon di Panduan EventBridge Pengguna Amazon.

Membuat pola acara khusus untuk EventBridge aturan

Anda dapat menemukan beberapa pola yang telah ditentukan dalam peristiwa sinkronisasi EventBridge for CloudFormation dan Git. Ini menyederhanakan bagaimana pola acara dibuat. Alih-alih menulis pola acara Anda sendiri, Anda dapat memilih nilai bidang pada formulir, dan EventBridge menghasilkan pola untuk Anda. Anda dapat membuat aturan baru menggunakan salah satu pola acara yang telah ditentukan ini atau membuat pola acara kustom Anda sendiri.

Saat layanan seperti CloudFormation mengirimkan acara ke bus peristiwa default, EventBridge gunakan pola acara yang ditentukan dalam aturan Anda untuk menentukan apakah acara tersebut harus dikirim ke target aturan. Pola peristiwa cocok dengan data dalam CloudFormation peristiwa yang diinginkan.

Setiap pola acara adalah objek JSON yang berisi:

- `sourceAtribut` yang mengidentifikasi layanan tempat acara berasal. Misalnya, `aws.cloudformation` atau `aws.codeconnections`.
- (Opsional): `detail-type` Atribut yang berisi array jenis acara yang cocok.
- (Opsional): `detail` Atribut yang berisi data acara lain yang cocok.

Misalnya, ID tumpukan, sumber daya yang terlibat, status berbagai sumber daya, dan data lain yang relevan dengan jenis peristiwa tertentu.

Misalnya, pola acara berikut cocok dengan semua peristiwa perubahan status sumber daya:

```
{
  "source": ["aws.cloudformation"],
  "detail-type": ["CloudFormation Resource Status Change"]
}
```

Sementara pola peristiwa berikut menggunakan data detail peristiwa untuk mencocokkan hanya peristiwa perubahan status sumber daya yang CloudFormation membuat baru `AWS::S3::Bucket` atau `AWS::SNS::Topic` sumber daya:

```
{
  "source": ["aws.cloudformation"],
  "detail-type": ["CloudFormation Resource Status Change"],
  "detail": {
    "status-details": {
      "status": ["CREATE_COMPLETE"]
    },
    "resource-type": ["AWS::S3::Bucket", "AWS::SNS::Topic"]
  }
}
```

Untuk informasi selengkapnya tentang penulisan pola peristiwa, lihat [Pola acara](#) di Panduan EventBridge Pengguna Amazon.

CloudFormation referensi detail acara

Semua peristiwa dari AWS layanan memiliki seperangkat bidang umum yang berisi metadata tentang acara tersebut, seperti AWS layanan yang menjadi sumber acara, waktu acara dibuat, akun dan wilayah tempat acara berlangsung, dan lainnya. Untuk definisi bidang umum ini, lihat [metadata acara AWS layanan](#) di Referensi AWS Acara.

Selain itu, setiap acara memiliki detail bidang yang berisi data khusus untuk peristiwa tertentu. Referensi di bawah ini mendefinisikan bidang detail untuk berbagai CloudFormation acara.

Saat menggunakan EventBridge untuk memilih dan mengelola CloudFormation acara, penting untuk mengingat hal berikut:

- `sourceBidang` menentukan sumber acara.

Misalnya, `aws.cloudformation` atau `aws.codeconnections`.

- `detail-typeBidang` menentukan jenis acara.

Misalnya, `CloudFormation Resource Status Change` atau `CloudFormation Drift Detection Status Change`.

- `detailBidang` berisi data yang spesifik untuk peristiwa tertentu.

Misalnya, ID tumpukan, sumber daya yang terlibat, status berbagai sumber daya, dan data lain yang relevan dengan jenis peristiwa tertentu.

Untuk informasi tentang membuat pola peristiwa yang memungkinkan aturan untuk mencocokkan CloudFormation peristiwa, lihat [Pola peristiwa](#) di Panduan EventBridge Pengguna Amazon.

Untuk informasi selengkapnya tentang peristiwa dan cara EventBridge memprosesnya, lihat [EventBridgeperistiwa](#) di Panduan EventBridge Pengguna Amazon.

Topik

- [Status Sumber Daya Ubah detail acara](#)
- [Stack Status Ubah detail acara](#)
- [Status Deteksi Drift Ubah detail acara](#)
- [StackSet Status Ubah detail acara](#)
- [StackSet Stack Instance Status Ubah detail acara](#)
- [StackSet Status Operasi Ubah detail acara](#)
- [Status Sinkronisasi Repositori Ubah detail acara](#)
- [Status Sinkronisasi Sumber Daya Ubah detail acara](#)

Status Sumber Daya Ubah detail acara

Di bawah ini adalah bidang detail untuk acara Perubahan Status Sumber Daya.

detail-typeBidang source dan disertakan karena mengandung nilai khusus untuk AWS CloudFormation acara.

```
{
  . . . ,
  "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

Mengidentifikasi jenis acara.

Untuk peristiwa status sumber daya, nilai ini adalah `CloudFormation Resource Status Change`.

source

Mengidentifikasi layanan yang menghasilkan peristiwa. Untuk CloudFormation acara, nilai ini adalah `aws.cloudformation`.

detail

Objek JSON yang berisi informasi tentang peristiwa. Layanan yang menghasilkan acara menentukan konten bidang ini.

Untuk peristiwa status sumber daya, data ini mencakup:

stack-id

ID tumpukan unik yang terkait dengan tumpukan.

logical-resource-id

Nama logis dari sumber daya seperti yang ditentukan dalam template.

physical-resource-id

Nama atau pengidentifikasi unik yang sesuai dengan ID instance fisik dari sumber daya yang didukung oleh CloudFormation.

status-details

status

Status sumber daya.

status-reason

Alasan status sumber daya.

resource-type

Jenis sumber daya. Misalnya, `AWS::S3::Bucket`.

client-request-token

Token akses yang digunakan untuk memanggil API. Semua peristiwa yang diprakarsai oleh operasi tumpukan tertentu diberi token permintaan klien yang sama, yang dapat Anda gunakan untuk melacak operasi. Operasi tumpukan yang dimulai dari konsol menggunakan format token `Console- StackOperation -ID`, yang membantu Anda mengidentifikasi operasi tumpukan dengan mudah. Misalnya, jika Anda membuat tumpukan menggunakan konsol, setiap kejadian tumpukan yang dihasilkan akan ditetapkan token yang sama dalam format berikut: `Console-CreateStack-7f59c3cf-00d2-40c7-b2ff-e75db0987002`.

Example Contoh: Acara Perubahan Status Sumber Daya

Berikut ini adalah contoh acara status sumber daya. Peristiwa ini merinci yang CloudFormation telah berhasil membuat sumber daya yang diminta, bucket Amazon S3, di tumpukan yang ditentukan.

```
{
  "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 Ubah detail acara

Di bawah ini adalah bidang detail untuk acara Stack Status Change.

`detail-type` Bidang source dan disertakan karena mengandung nilai khusus untuk AWS CloudFormation acara.

```

{
    . . . ,
    "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

Mengidentifikasi jenis acara.

Untuk peristiwa status tumpukan, nilai ini adalah `CloudFormation Stack Status Change`.

source

Mengidentifikasi layanan yang menghasilkan peristiwa. Untuk CloudFormation acara, nilai ini adalah `aws.cloudformation`.

detail

Objek JSON yang berisi informasi tentang peristiwa. Layanan yang menghasilkan acara menentukan konten bidang ini.

Untuk peristiwa status tumpukan, data ini mencakup:

`stack-id`

ID tumpukan unik yang terkait dengan tumpukan.

`status-details`

`status`

Status tumpukan.

Untuk daftar lengkap kode status tumpukan, lihat [Kode status tumpukan](#).

`status-reason`

Alasan status sumber daya.

`client-request-token`

Token akses yang digunakan untuk memanggil API. Semua peristiwa yang diprakarsai oleh operasi tumpukan tertentu diberi token permintaan klien yang sama, yang dapat Anda gunakan untuk melacak operasi. Operasi tumpukan yang dimulai dari konsol menggunakan format token Console- StackOperation -ID, yang membantu Anda mengidentifikasi operasi tumpukan dengan mudah. Misalnya, jika Anda membuat tumpukan menggunakan konsol, setiap kejadian tumpukan yang dihasilkan akan ditetapkan token yang sama dalam format berikut: `Console-CreateStack-7f59c3cf-00d2-40c7-b2ff-e75db0987002`.

Example Contoh: Acara Stack Status

Berikut ini adalah contoh peristiwa status tumpukan, di mana CloudFormation telah berhasil membuat tumpukan yang diminta, `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":""
    }
  }
}

```

Status Deteksi Drift Ubah detail acara

Di bawah ini adalah bidang detail untuk peristiwa deteksi drift tumpukan.

`detail-type` Bidang source dan disertakan karena mengandung nilai khusus untuk AWS CloudFormation acara.

```

{
  . . . ,
  "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

Mengidentifikasi jenis acara.

Untuk peristiwa deteksi stack drift, nilai ini adalah `CloudFormation Drift Detection Status Change`.

source

Mengidentifikasi layanan yang menghasilkan peristiwa. Untuk CloudFormation acara, nilai ini adalah `aws.cloudformation`.

detail

Objek JSON yang berisi informasi tentang peristiwa. Layanan yang menghasilkan acara menentukan konten bidang ini.

Untuk kejadian deteksi stack drift, data ini mencakup:

stack-id

ID tumpukan unik yang terkait dengan tumpukan.

stack-drift-detection-id

Id deteksi drift tumpukan.

status-details

stack-drift-status

Status drift tumpukan.

detection-status

Status operasi deteksi drift.

drift-detection-details

drifted-stack-resource-count

Jumlah sumber daya melayang. Ketika nilainya -1, deteksi drift sedang berlangsung.

Semua bilangan bulat non-negatif lainnya mewakili jumlah sebenarnya dari sumber daya yang hanyut.

client-request-token

Token akses yang digunakan untuk memanggil API. Semua peristiwa yang diprakarsai oleh operasi tumpukan tertentu diberi token permintaan klien yang sama, yang dapat Anda gunakan untuk melacak operasi. Operasi tumpukan yang dimulai dari konsol menggunakan format token `Console-StackOperation-ID`, yang membantu Anda mengidentifikasi operasi tumpukan dengan mudah. Misalnya, jika Anda membuat tumpukan menggunakan konsol, setiap kejadian tumpukan yang dihasilkan akan ditetapkan token yang sama dalam format berikut: `Console-CreateStack-7f59c3cf-00d2-40c7-b2ff-e75db0987002`.

Example Contoh: Acara Stack Drift Detection

Berikut ini adalah contoh peristiwa Stack Drift Detection. Peristiwa ini merinci yang CloudFormation telah menyelesaikan deteksi drift pada tumpukan yang ditentukan, dan bahwa tumpukan saat ini memiliki status drift DRIFTED karena satu sumber daya yang hanyut.

```
{
  "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 Status Ubah detail acara

Di bawah ini adalah bidang detail untuk acara Perubahan StackSet Status.

`detail-type` Bidang `source` dan disertakan karena mengandung nilai khusus untuk AWS CloudFormation acara.

```
{
  . . . ,
  "detail-type": "CloudFormation StackSet Status Change",
  "source": "aws.cloudformation",
  . . . ,
  "detail": {
```

```
"stack-set-arn" : "string",
"status-details": {
  "status": "string"
}
}
```

detail-type

Mengidentifikasi jenis acara.

Untuk acara acara StackSet status, nilai ini adalah CloudFormation StackSet Status Change.

source

Mengidentifikasi layanan yang menghasilkan peristiwa. Untuk CloudFormation acara, nilai ini adalah `aws.cloudformation`.

detail

Objek JSON yang berisi informasi tentang peristiwa. Layanan yang menghasilkan acara menentukan konten bidang ini.

Untuk acara acara StackSet status, data ini mencakup:

stack-set-arn

Nama Sumber Daya Amazon (ARN) yang terkait dengan kumpulan tumpukan.

status-details

status

StackSet Statusnya.

Nilai yang valid: ACTIVE | DELETED

Example Contoh: Acara Perubahan StackSet Status

Berikut ini adalah contoh peristiwa Perubahan StackSet Status. Rincian peristiwa ini yang CloudFormation telah menghapus kumpulan tumpukan yang ditentukan.

```
{
```

```

"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 Stack Instance Status Ubah detail acara

Di bawah ini adalah bidang detail untuk peristiwa status instance StackSet stack.

`detail-type` Bidang `source` dan disertakan karena mengandung nilai khusus untuk AWS CloudFormation acara.

```

{
  . . . ,
  "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

Mengidentifikasi jenis acara.

Untuk kejadian status instance StackSet stack, nilai ini adalah `CloudFormation StackSet StackInstance Status Change`.

source

Mengidentifikasi layanan yang menghasilkan peristiwa. Untuk CloudFormation acara, nilai ini adalah `aws.cloudformation`.

detail

Objek JSON yang berisi informasi tentang peristiwa. Layanan yang menghasilkan acara menentukan konten bidang ini.

Untuk kejadian status instance StackSet stack, data ini mencakup:

stack-set-arn

Nama Sumber Daya Amazon (ARN) yang terkait dengan file. StackSet

stack-id

ID tumpukan unik yang terkait dengan instance stack.

action

Jenis operasi stack set.

Nilai yang valid: `CREATE` | `UPDATE` | `DELETE` | `DETECT_DRIFT`

status-details

status

Status StackSet instance.

Untuk detail selengkapnya, lihat [Kode status instance tumpukan](#).

Nilai yang valid: `CURRENT` | `OUTDATED` | `INOPERABLE`

status-reason

Alasan status StackSet instance.

detailed-status

StackSet Contoh rinci status rinci.

Nilai yang valid: CANCELLED FAILED | FAILED_IMPORT | INOPERABLE | PENDING | RUNNING | SKIPPED_SUSPENDED_ACCOUNT | SUCCEEDED

Example Contoh: Acara Perubahan Status Instans StackSet Stack

Berikut ini adalah contoh StackSet Stack Instance Status Change event.

```
{
  "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 Status Operasi Ubah detail acara

Di bawah ini adalah bidang detail untuk peristiwa Perubahan Status StackSet Operasi.

detail-typeBidang source dan disertakan karena mengandung nilai khusus untuk AWS CloudFormation acara.

```
{
  . . . ,
  "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

Mengidentifikasi jenis acara.

Untuk peristiwa status StackSet operasi, nilai ini adalah `CloudFormation StackSet Operation Status Change`.

source

Mengidentifikasi layanan yang menghasilkan peristiwa. Untuk CloudFormation acara, nilai ini adalah `aws.cloudformation`.

detail

Objek JSON yang berisi informasi tentang peristiwa. Layanan yang menghasilkan acara menentukan konten bidang ini.

Untuk peristiwa status StackSet operasi, data ini mencakup:

stack-set-arn

Nama Sumber Daya Amazon (ARN) yang terkait dengan file. StackSet

stack-set-operation-id

ID unik yang terkait dengan StackSet operasi.

status-details

status

Status StackSet operasi.

Untuk detail selengkapnya, lihat [StackSets kode status](#).

Nilai yang valid: RUNNING SUCCEEDED | FAILED | STOPPING | STOPPED | QUEUED

Example Contoh: Acara Perubahan Status StackSet Operasi

Berikut ini adalah contoh event StackSet Operation Status Change. Rincian peristiwa yang CloudFormation telah berhasil menyelesaikan operasi yang diminta pada set tumpukan yang ditentukan.

```
{
  "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"
    }
  }
}
```

Status Sinkronisasi Repositori Ubah detail acara

Di bawah ini adalah bidang detail untuk peristiwa Perubahan Status Sinkronisasi Repositori.

`detail-type` Bidang source dan disertakan karena mengandung nilai khusus untuk AWS CloudFormation acara.

```
{
  . . . ,
  "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

Mengidentifikasi jenis acara.

Untuk peristiwa status Sinkronisasi Repositori, nilai ini adalah. Git Sync Repository Sync Status Change

source

Mengidentifikasi layanan yang menghasilkan peristiwa. Untuk peristiwa sinkronisasi Git, nilai ini adalah `aws.codeconnections`.

detail

Objek JSON yang berisi informasi tentang peristiwa. Layanan yang menghasilkan acara menentukan konten bidang ini.

Untuk peristiwa status sinkronisasi Repositori, data ini mencakup:

connectionArn

Nama Sumber Daya Amazon (ARN) yang terkait dengan. CodeConnections

providerType

Penyedia Git terhubung ke CloudFormation.

Nilai yang valid: GitHub | GitHub Enterprise | GitLab | BitBucket

`repositoryName`

Nama repositori Git.

`repositoryArn`

ARN terkait dengan repositori Git.

`repositoryLinkId`

ID unik yang terkait dengan tautan repositori.

`ownerId`

ID unik yang terkait dengan pemilik repositori.

`commit`

ID unik yang terkait dengan komit repositori.

`branch`

ID unik yang terkait dengan cabang repositori.

`syncType`

Jenis sinkronisasi yang sedang dilakukan.

`status`

Status sinkronisasi repositori saat ini.

Nilai yang valid: FAILED | INITIATED | IN_PROGRESS | SUCCEEDED

`previousSync`

Status sinkronisasi sebelum status saat ini.

Nilai yang valid: FAILED | INITIATED | IN_PROGRESS | SUCCEEDED

Example Contoh: Acara Perubahan Status Sinkronisasi Repositori

Berikut ini adalah contoh peristiwa Perubahan Status Sinkronisasi Repositori. Detail acara yang CodeConnections telah berhasil menyinkronkan repositori.

```
{
```

```

"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",
}
}

```

Status Sinkronisasi Sumber Daya Ubah detail acara

Di bawah ini adalah bidang detail untuk acara Perubahan Status Sinkronisasi Sumber Daya.

`detail-type` Bidang `source` dan disertakan karena mengandung nilai khusus untuk AWS CloudFormation acara.

```

{
  . . . ,
  "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

Mengidentifikasi jenis acara.

Untuk peristiwa status Sinkronisasi Repositori, nilai ini adalah. Git Sync Repository Sync Status Change

source

Mengidentifikasi layanan yang menghasilkan peristiwa. Untuk peristiwa sinkronisasi Git, nilai ini adalah `aws.codeconnections`.

detail

Objek JSON yang berisi informasi tentang peristiwa. Layanan yang menghasilkan acara menentukan konten bidang ini.

Untuk peristiwa status sinkronisasi sumber daya, data ini mencakup:

providerType

Penyedia Git terhubung ke CloudFormation.

Nilai yang valid: GitHub | GitHub Enterprise | GitLab | BitBucket

commit

ID unik yang terkait dengan komit repositori.

repositoryName

Nama repositori Git.

branch

ID unik yang terkait dengan cabang repositori.

syncType

Jenis sinkronisasi yang sedang dilakukan.

syncTarget

Tumpukan target untuk sinkronisasi sumber daya.

status

Status sinkronisasi repositori saat ini.

Nilai yang valid: FAILED | INITIATED | IN_PROGRESS | SUCCEEDED

previousSync

Status sinkronisasi sebelum status saat ini.

Nilai yang valid: FAILED | INITIATED | IN_PROGRESS | SUCCEEDED

Example Contoh: Acara Perubahan Status Sinkronisasi Sumber Daya

Berikut ini adalah contoh peristiwa perubahan status sinkronisasi sumber daya. Detail acara yang CodeConnections telah berhasil menyinkronkan sumber daya.

```
{
  "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"
  }
}
```

Memahami CloudFormation kuota

Anda Akun AWS memiliki CloudFormation kuota yang mungkin perlu Anda ketahui saat membuat templat dan membuat tumpukan. Dengan memahami kuota ini, Anda dapat menghindari kesalahan pembatasan yang mengharuskan Anda mendesain ulang templat atau tumpukan.

Tabel berikut menunjukkan CloudFormation kuota.

Kuota	Deskripsi	Nilai	Strategi penyetelan
cfn-signal tunggu data kondisi	Jumlah maksimum data yang cfn-signal dapat dilewati.	4.096 byte	Untuk meneruskan jumlah yang lebih besar, kirim data ke bucket Amazon S3, lalu gunakan cfn-signal untuk meneruskan URL Amazon S3 ke bucket itu.
Respon sumber daya khusus	Jumlah maksimum data yang dapat dilewati oleh penyedia sumber daya khusus.	4.096 byte	
Referensi dinamis per template	Jumlah maksimum referensi dinamis yang diizinkan dalam satu template CloudFormation tumpukan.	60 referensi dinamis dalam template tumpukan	
Kait per akun	Jumlah maksimum Hooks yang diizinkan per akun per Wilayah.	100 kait	
Kait per sumber daya	Jumlah maksimum Hooks yang dapat	100 kait	

Kuota	Deskripsi	Nilai	Strategi penyetelan
	dikonfigurasi per sumber daya.		
Ukuran konfigurasi kait	Jumlah maksimum data konfigurasi yang dapat disimpan oleh Hook.	204,8 KB	
Pemetaan	Jumlah maksimum pemetaan yang dapat Anda deklarasikan dalam template Anda. CloudFormation	200 pemetaan	Untuk menentukan lebih banyak pemetaan, pisahkan templat ke dalam beberapa template menggunakan, misalnya, tumpukan yang di-nest .
Atribut pemetaan	Jumlah maksimum atribut pemetaan untuk setiap pemetaan yang dapat Anda deklarasikan dalam template Anda. CloudFormation	200 atribut	Untuk menentukan lebih banyak atribut pemetaan, pisahkan atribut menjadi beberapa pemetaan.
Nama pemetaan dan pemetaan nama atribut	Ukuran maksimum setiap nama pemetaan.	255 karakter	
Modul	Jumlah maksimum modul yang dapat Anda daftarkan di CloudFormation registri, per akun dan Wilayah.	100 modul	

Kuota	Deskripsi	Nilai	Strategi penyetelan
Versi modul	Jumlah maksimum versi yang dapat Anda daftarkan di CloudFormation registri untuk modul yang diberikan.	100 versi	Untuk mendaftarkan versi baru, pertama-tama gunakan DeregisterType untuk membatalkan pendaftaran versi yang tidak Anda gunakan lagi.
Tumpukan bersarang	Jumlah maksimum CloudFormation sumber daya yang dapat dibuat, diperbarui, atau dihapus oleh tumpukan bersarang per operasi. Misalnya, Anda dapat memiliki hierarki tumpukan bersarang dengan lebih dari 2500 total sumber daya, tetapi Anda tidak dapat membuat, memperbarui, atau menghapus lebih dari 2500 sumber daya tersebut dalam satu penerapan.	2500 sumber daya	Pisahkan hierarki tumpukan menjadi tumpukan yang berbeda.

Kuota	Deskripsi	Nilai	Strategi penyetelan
Keluaran	Jumlah maksimum output yang dapat Anda deklarasikan dalam template Anda. CloudFormation	200 output	
Nama keluaran	Ukuran maksimum dari nama output.	255 karakter	
Parameter	Jumlah maksimum parameter yang dapat Anda deklarasikan dalam template Anda CloudFormation.	200 parameter	Untuk menentukan lebih banyak parameter, Anda dapat menggunakan pemetaan atau daftar untuk menetapkan beberapa nilai untuk parameter tunggal.
Nama parameter	Ukuran maksimum nama parameter.	255 karakter	
Nilai parameter	Ukuran maksimum dari nilai parameter.	4.096 byte	Untuk menggunakan nilai parameter yang lebih besar, buat beberapa parameter dan kemudian gunakan Fn::Join fungsi untuk menggabungkan beberapa nilai menjadi satu nilai.

Kuota	Deskripsi	Nilai	Strategi penyetelan
Sumber daya pribadi	Jumlah maksimum sumber daya pribadi yang dapat Anda daftarkan di CloudFormation registri per akun, per Wilayah.	50 sumber daya privat	
Versi sumber daya pribadi	Jumlah maksimum versi yang dapat Anda daftarkan di CloudFormation registri untuk sumber daya pribadi tertentu.	50 sumber daya privat	Untuk mendaftarkan versi baru, pertama-tama gunakan DeregisterType untuk membatalkan pendaftaran versi yang tidak Anda gunakan lagi.
Sumber Daya	Jumlah maksimum sumber daya yang dapat Anda deklarasikan dalam template Anda CloudFormation .	500 sumber daya	Untuk menentukan lebih banyak sumber daya, pisahkan templat Anda ke dalam beberapa templat dengan menggunakan, misalnya, tumpukan yang di-nest .

Kuota	Deskripsi	Nilai	Strategi penyetelan
Sumber daya dalam operasi tumpukan bersamaan	Jumlah maksimum sumber daya yang dapat Anda libatkan dalam operasi tumpukan (membuat, memperbarui, atau menghapus operasi) di Wilayah Anda pada waktu tertentu.	Gunakan DescribeAccountLimits API untuk menentukan batas saat ini untuk akun di Wilayah tertentu.	
Nama sumber daya	Ukuran maksimum nama sumber daya.	255 karakter	
Tumpukan	Jumlah maksimum CloudFormation tumpukan yang dapat Anda buat.	2000 tumpukan	Untuk membuat lebih banyak tumpukan, hapus tumpukan yang tidak Anda perlukan atau minta peningkatan jumlah tumpukan maksimum di tumpukan Anda. Akun AWS Untuk informasi selengkapnya, lihat kuota layanan AWS di Referensi Umum AWS.
Nama tumpukan	Ukuran maksimum nama tumpukan.	128 karakter	

Kuota	Deskripsi	Nilai	Strategi penyetelan
StackSets	Jumlah maksimum set CloudFormation tumpukan yang dapat Anda buat di akun administrator Anda.	1000 set tumpukan	Untuk membuat lebih banyak set tumpukan, hapus kumpulan tumpukan yang tidak Anda perlukan atau minta peningkatan jumlah set tumpukan maksimum di set tumpukan Anda Akun AWS. Untuk informasi selengkapnya, lihat Kuota layanan AWS di Referensi Umum AWS.
Contoh tumpukan	Jumlah maksimum instance tumpukan yang dapat Anda buat per set tumpukan.	100.000 instance tumpukan per set tumpukan	Untuk membuat lebih banyak instance tumpukan, hapus instance tumpukan yang tidak Anda perlukan atau minta peningkatan jumlah maksimum instance tumpukan di. Akun AWS Untuk informasi selengkapnya, lihat kuota layanan AWS di Referensi Umum AWS.

Kuota	Deskripsi	Nilai	Strategi penyetelan
StackSets operasi contoh	Jumlah maksimum instance tumpukan, di semua set tumpukan, yang dapat Anda jalankan operasi di setiap Wilayah secara bersamaan, per akun administrator.	10.000 operasi	Batas ini berlaku di semua set tumpukan yang terlibat di sebuah Wilayah. Ini termasuk instans tumpukan yang dipengaruhi oleh set tumpukan operasi pembuatan dan pembaruan, serta membuat, memperbarui, atau menghapus instans tumpukan secara langsung.
StackSets operasi antrian	Jumlah maksimum operasi antrian untuk tumpukan yang ditetapkan pada waktu tertentu.	10.000 operasi	
Tumpukan yang diimpor menggunakan objek S3 per operasi set tumpukan	Jumlah maksimum tumpukan yang dapat Anda impor menggunakan objek S3 per operasi set tumpukan.	200 tumpukan	
Tumpukan yang diimpor menggunakan id tumpukan sebaris per operasi set tumpukan	Jumlah maksimum tumpukan yang dapat Anda impor menggunakan tumpukan sebaris IDs per operasi set tumpukan.	10 tumpukan	

Kuota	Deskripsi	Nilai	Strategi penyetelan
Ukuran tubuh template dalam permintaan	Ukuran maksimum badan templat yang dapat Anda lewatkan dalam <code>CreateStack</code> , <code>UpdateStack</code> , atau <code>ValidateTemplate</code> permintaan.	51.200 byte	Untuk menggunakan isi templat yang lebih besar, pisahkan templat ke dalam beberapa templat menggunakan, misalnya, tumpukan yang di-nest . Atau unggah template ke bucket Amazon S3.
Ukuran tubuh template dalam objek Amazon S3	Ukuran maksimum badan template yang dapat Anda lewatkan dalam objek Amazon S3 untuk <code>ValidateTemplate</code> permintaan <code>CreateStack</code> , <code>UpdateStack</code> , dengan URL template Amazon S3.	1 MB	Untuk menggunakan isi templat yang lebih besar, pisahkan templat ke dalam beberapa templat dengan menggunakan, misalnya, tumpukan yang di-nest . Atau gunakan minifikasi untuk mengurangi ukuran CloudFormation template.
Deskripsi template	Ukuran maksimum deskripsi template.	1.024 byte	
Versi per kait	Jumlah maksimum versi yang dapat Anda buat per Hook.	100 versi	

Ketersediaan fitur

Tidak semua fitur CloudFormation mungkin tersedia di setiap Wilayah. Untuk informasi selengkapnya Wilayah AWS, lihat [tabel Wilayah infrastruktur global](#).

- [Makro](#) saat ini tidak tersedia di Wilayah berikut:
 - Asia Pasifik (Jakarta)
- [Melakukan blue/green penerapan ECS melalui CodeDeploy penggunaan CloudFormation](#) saat ini tidak tersedia di Wilayah berikut:
 - Afrika (Cape Town)
 - Asia Pasifik (Osaka)
 - Eropa (Milan)

StackSets dan makro

StackSets saat ini tidak mendukung pembuatan atau pembaruan kumpulan tumpukan dengan izin yang dikelola layanan dari templat yang berisi makro. Ini termasuk transformasi, yang merupakan makro yang dihosting oleh CloudFormation. Untuk informasi selengkapnya tentang makro, lihat [Makro templat](#).

Pemecahan masalah CloudFormation

Saat Anda menggunakan CloudFormation, Anda mungkin mengalami masalah saat membuat, memperbarui, atau menghapus CloudFormation tumpukan. Bagian berikut ini dapat membantu Anda memecahkan beberapa masalah umum yang mungkin Anda temui.

Untuk pertanyaan umum tentang CloudFormation, lihat [AWS CloudFormation FAQs](#). Anda juga dapat mencari jawaban dan memposting pertanyaan di [CloudFormation komunitas](#) di AWS re:Post.

Topik

- [Panduan pemecahan masalah](#)
- [Memecahkan masalah kesalahan](#)
- [Menghubungi dukungan](#)

Panduan pemecahan masalah

Jika CloudFormation gagal membuat, memperbarui, atau menghapus tumpukan, Anda dapat melihat pesan kesalahan atau log untuk membantu Anda mempelajari lebih lanjut tentang masalah ini. Tugas-tugas berikut menjelaskan metode umum untuk memecahkan masalah. CloudFormation Untuk informasi tentang kesalahan dan solusi tertentu, lihat bagian [Memecahkan masalah kesalahan](#).

- Gunakan [konsol CloudFormation](#) untuk melihat status tumpukan Anda. Di konsol, Anda dapat melihat daftar peristiwa tumpukan saat tumpukan Anda sedang dibuat, diperbarui, atau dihapus. Dari daftar ini, temukan peristiwa kegagalan dan kemudian lihat alasan status untuk peristiwa tersebut. Alasan status mungkin berisi pesan kesalahan dari CloudFormation atau dari layanan tertentu yang dapat membantu Anda memecahkan masalah Anda. Untuk informasi selengkapnya tentang melihat peristiwa tumpukan, lihat [Lihat acara CloudFormation tumpukan](#).
- Untuk EC2 masalah Amazon, lihat log `cloud-init` dan `cfn`. Log ini dipublikasikan di EC2 instance Amazon di `/var/log/` direktori. Log ini menangkap proses dan output perintah saat CloudFormation menyiapkan instance Anda. Untuk Windows, lihat layanan EC2 Configure in%ProgramFiles%\Amazon\EC2ConfigService, EC2 Launch in%ProgramData%\Amazon\EC2-Windows\Launch\Logs, EC2 Launch v2 in%ProgramData%\Amazon\EC2Launch\log, dan cfn log inC:\cfn\log.

Anda juga dapat mengonfigurasi CloudFormation template Anda sehingga log dipublikasikan ke Amazon CloudWatch, yang menampilkan log di AWS Management Console sehingga Anda

tidak perlu terhubung ke EC2 instans Amazon Anda. Untuk informasi selengkapnya, [lihat Melihat CloudFormation log di konsol](#) di Blog Manajemen Aplikasi.

Memecahkan masalah kesalahan

Ketika Anda menemukan kesalahan berikut dengan CloudFormation tumpukan Anda, Anda dapat menggunakan solusi berikut untuk membantu Anda menemukan sumber masalah dan memperbaikinya.

Topik

- [Gagal menghapus tumpukan](#)
- [Kesalahan dependensi](#)
- [AWS Config dan AWS Systems Manager konflik](#)
- [Kesalahan penguraian parameter saat melewati daftar](#)
- [Izin IAM tidak mencukupi](#)
- [Nilai tidak valid atau properti sumber daya tidak didukung](#)
- [Kuota terlampaui](#)
- [Tumpukan nested terjebak dalam UPDATE_COMPLETE_CLEANUP_IN_PROGRESS, UPDATE_ROLLBACK_COMPLETE_CLEANUP_IN_PROGRESS, atau UPDATE_ROLLBACK_IN_PROGRESS](#)
- [Tidak ada pembaruan untuk dilakukan](#)
- [Sumber daya gagal distabilkan selama membuat, memperbarui, atau menghapus operasi tumpukan](#)
- [Grup keamanan tidak ada di VPC](#)
- [Pembaruan rollback gagal](#)
- [Kondisi tunggu tidak menerima jumlah sinyal yang diperlukan dari EC2 instans Amazon](#)
- [Sumber daya dihilangkan dari tumpukan tetapi tidak dihapus](#)

Gagal menghapus tumpukan

Untuk mengatasi situasi ini, coba yang berikut ini:

- Beberapa sumber daya harus kosong sebelum dapat dihapus. Misalnya, Anda harus menghapus semua objek di bucket Amazon S3 atau menghapus semua instance di grup EC2 keamanan Amazon sebelum dapat menghapus bucket atau grup keamanan.
- Pastikan Anda memiliki izin IAM yang diperlukan untuk menghapus sumber daya di tumpukan. Selain CloudFormation izin, Anda harus diizinkan untuk menggunakan layanan yang mendasarinya, seperti Amazon S3 atau Amazon. EC2
- Ketika tumpukan berada dalam DELETE_FAILED status karena CloudFormation tidak dapat menghapus sumber daya, jalankan kembali [penghapusan](#) dengan RetainResources parameter dan tentukan sumber daya yang CloudFormation tidak dapat dihapus. CloudFormation menghapus tumpukan tanpa menghapus sumber daya yang dipertahankan. Mempertahankan sumber daya berguna saat Anda tidak dapat menghapus sumber daya, seperti bucket S3 yang berisi objek yang ingin Anda simpan, tetapi Anda masih ingin menghapus tumpukan.

Setelah menghapus tumpukan, Anda dapat menghapus sumber daya yang disimpan secara manual dengan menggunakan AWS layanan terkaitnya.

Atau, Anda dapat mempertimbangkan untuk menggunakan FORCE_DELETE_STACK opsi dengan DeletionMode parameter. Untuk informasi selengkapnya tentang penghapusan paksa tumpukan, lihat [Hapus tumpukan dari CloudFormation konsol](#).

- Anda tidak dapat menghapus tumpukan yang mengaktifkan perlindungan penghentian. Jika Anda mencoba menghapus tumpukan dengan perlindungan penghentian diaktifkan, penghapusan gagal dan tumpukan--termasuk statusnya--tetap tidak berubah. Nonaktifkan perlindungan penghentian pada tumpukan, lalu lakukan operasi penghapusan lagi.

Ini mencakup [tumpukan nested](#) yang tumpukan akarnya memiliki perlindungan penghentian yang aktif. Nonaktifkan perlindungan terminasi pada tumpukan root, lalu lakukan operasi hapus lagi. Sangat disarankan agar Anda tidak menghapus tumpukan nested secara langsung, tetapi hanya menghapusnya sebagai bagian dari menghapus tumpukan akar dan semua sumber dayanya.

Untuk informasi selengkapnya, lihat [Lindungi CloudFormation tumpukan agar tidak dihapus](#).

- Untuk semua masalah lain, jika ada AWS Dukungan, Anda dapat membuat Dukungan kasus. Lihat [Menghubungi dukungan](#).

Kesalahan dependensi

Untuk mengatasi kesalahan dependensi, tambahkan atribut DependsOn ke sumber daya yang bergantung pada sumber daya lain dalam templat Anda. Dalam beberapa kasus, Anda harus secara

eksplisit mendeklarasikan dependensi sehingga CloudFormation dapat membuat atau menghapus sumber daya dalam urutan yang benar. Misalnya, jika Anda membuat IP Elastis dan VPC dengan gateway Internet di tumpukan yang sama, IP Elastis harus bergantung pada lampiran gateway Internet. Untuk informasi lebih lanjut, lihat [Atribut DependsOn](#).

AWS Config dan AWS Systems Manager konflik

AWS Config dan AWS Systems Manager dapat mengotomatiskan tugas manajemen infrastruktur yang dapat menyebabkan konflik dengan penyebaran tumpukan. CloudFormation Lakukan hal berikut untuk menghindari potensi konflik:

- Tinjau konfigurasi AWS Config dan Systems Manager yang terkait Akun AWS dan Wilayah AWS.
- Periksa aturan aktif atau dokumen otomatisasi yang mungkin dipicu selama CloudFormation penerapan. Ini berpotensi menyebabkan konflik atau dependensi sumber daya yang bertentangan dengan penerapan Anda.
- Periksa CloudFormation template Anda untuk sumber daya apa pun yang dikelola oleh AWS Config dan Systems Manager. Periksa potensi tumpang tindih atau saling ketergantungan dan pertimbangkan untuk menyesuaikan templat atau konfigurasi otomatisasi untuk menghindari konflik.
- Menonaktifkan atau menangguhkan sementara AWS Config aturan terkait atau otomatisasi Systems Manager selama CloudFormation penerapan. Ingatlah untuk mengembalikan konfigurasi asli setelah penerapan berhasil untuk mempertahankan tingkat otomatisasi dan kepatuhan yang diinginkan.
- Tinjau CloudFormation log dan pesan kesalahan untuk referensi apa pun AWS Config dan masalah terkait Manajer Sistem untuk membantu menentukan sumber konflik.

Untuk informasi selengkapnya tentang AWS Config aturan, lihat [Mengevaluasi Sumber Daya dengan Aturan AWS Config](#).

Untuk informasi selengkapnya tentang otomatisasi Systems Manager, lihat [AWS Systems Manager Otomatisasi](#).

Kesalahan penguraian parameter saat melewati daftar

Saat Anda menggunakan AWS CLI or CloudFormation untuk meneruskan daftar, tambahkan karakter escape (\) sebelum setiap koma. Contoh berikut menunjukkan bagaimana Anda menentukan parameter input saat menggunakan AWS CLI.

```
ParameterKey=CIDR,ParameterValue='10.10.0.0/16\,10.10.0.0/24\,10.10.1.0/24'
```

Izin IAM tidak mencukupi

Ketika Anda bekerja dengan CloudFormation tumpukan, Anda tidak hanya memerlukan izin untuk digunakan CloudFormation, Anda juga harus memiliki izin untuk menggunakan layanan dasar yang dijelaskan dalam template Anda. Misalnya, jika Anda membuat bucket Amazon S3 atau memulai EC2 instans Amazon, Anda memerlukan izin ke Amazon S3 atau Amazon. EC2 Tinjau kebijakan IAM Anda dan verifikasi bahwa Anda memiliki izin yang diperlukan sebelum bekerja dengan CloudFormation tumpukan. Untuk informasi selengkapnya, lihat [Kontrol CloudFormation akses dengan AWS Identity and Access Management](#).

Nilai tidak valid atau properti sumber daya tidak didukung

Saat Anda membuat atau memperbarui CloudFormation tumpukan, tumpukan Anda dapat gagal karena parameter input yang tidak valid, nama properti sumber daya yang tidak didukung, atau nilai properti sumber daya yang tidak didukung. Untuk parameter input, verifikasi bahwa sumber daya ada. Misalnya, saat Anda menentukan EC2 key pair Amazon atau ID VPC, sumber daya harus ada di akun Anda dan di wilayah tempat Anda membuat atau memperbarui tumpukan Anda. Anda dapat menggunakan [tipe parameter AWS-specific](#) untuk memastikan bahwa Anda menggunakan nilai yang valid.

Untuk nama dan nilai properti sumber daya, perbarui templat Anda untuk menggunakan nama dan nilai yang valid. Untuk informasi selengkapnya, lihat [referensi jenis AWS sumber daya dan properti](#).

Kuota terlampaui

Verifikasi bahwa Anda tidak mencapai kuota sumber daya. Misalnya, jumlah maksimum default instans Amazon EC2 On-Demand yang dapat Anda luncurkan adalah 5. Jika mencoba membuat lebih banyak instans Amazon EC2 On-Demand daripada kuota akun Anda, pembuatan instans gagal dan Anda menerima kesalahan. Status=start_failed Untuk melihat AWS kuota default berdasarkan layanan, lihat [kuota AWS layanan](#) di. Referensi Umum AWS

Untuk CloudFormation kuota dan strategi penyesuaian, lihat. [Memahami CloudFormation kuota](#)

Juga, selama pembaruan, jika sumber daya diganti, CloudFormation buat sumber daya baru sebelum menghapus yang lama. Penggantian ini dapat menempatkan akun Anda di atas kuota sumber daya, yang akan menyebabkan pembaruan Anda gagal. Anda dapat menghapus kelebihan sumber daya atau meminta [peningkatan kuota](#).

Tumpukan nested terjebak dalam

UPDATE_COMPLETE_CLEANUP_IN_PROGRESS,
UPDATE_ROLLBACK_COMPLETE_CLEANUP_IN_PROGRESS, atau
UPDATE_ROLLBACK_IN_PROGRESS

Tumpukan nested gagal untuk melakukan rollback. Karena potensi dependensi sumber daya antara tumpukan bersarang, CloudFormation tidak mulai membersihkan sumber daya tumpukan bersarang sampai semua tumpukan bersarang telah diperbarui atau telah diputar kembali. Ketika tumpukan bersarang gagal diputar kembali, CloudFormation membatalkan semua operasi, terlepas dari status tumpukan bersarang lainnya. Tumpukan bersarang yang selesai memperbarui atau memutar kembali tetapi tidak menerima sinyal dari CloudFormation untuk mulai membersihkan karena nested lain gagal untuk memutar kembali dalam keadaan **UPDATE_COMPLETE_CLEANUP_IN_PROGRESS** atau **UPDATE_ROLLBACK_COMPLETE_CLEANUP_IN_PROGRESS**. Tumpukan nested yang gagal diperbarui tetapi tidak menerima sinyal untuk mulai melakukan rollback berada dalam status **UPDATE_ROLLBACK_IN_PROGRESS**.

Tumpukan bersarang mungkin gagal diputar kembali karena perubahan yang dibuat di luar CloudFormation, ketika template tumpukan tidak secara akurat mencerminkan status tumpukan. Tumpukan bersarang mungkin juga gagal jika grup Auto Scaling di tumpukan bersarang memiliki periode batas waktu sinyal sumber daya yang tidak mencukupi saat grup dibuat atau diperbarui.

Untuk memperbaiki tumpukan, hubungi [AWS Dukungan](#).

Tidak ada pembaruan untuk dilakukan

Untuk memperbarui CloudFormation tumpukan, Anda harus mengirimkan perubahan nilai template atau parameter ke CloudFormation. Namun, CloudFormation tidak akan mengenali beberapa perubahan template sebagai pembaruan, seperti perubahan pada kebijakan penghapusan, kebijakan pembaruan, deklarasi kondisi, atau deklarasi keluaran. Jika Anda perlu membuat perubahan tersebut tanpa membuat perubahan lain, Anda dapat menambahkan atau mengubah atribut metadata untuk sumber daya mana pun. Atribut metadata dapat berupa nilai arbitrer, karena CloudFormation tidak menafsirkan kontennya. Untuk informasi lebih lanjut, lihat [Atribut Metadata](#).

Sumber daya gagal distabilkan selama membuat, memperbarui, atau menghapus operasi tumpukan

Sumber daya tidak merespons karena operasi melebihi periode CloudFormation batas waktu atau AWS layanan terganggu. Untuk gangguan layanan, [periksa apakah](#) AWS layanan yang relevan sedang berjalan, lalu coba lagi operasi tumpukan.

Jika AWS layanan telah berhasil berjalan, periksa apakah tumpukan Anda berisi salah satu sumber daya berikut:

- `AWS::AutoScaling::AutoScalingGroup` untuk membuat, memperbarui, dan menghapus operasi
- `AWS::CertificateManager::Certificate` untuk membuat operasi
- `AWS::CloudFormation::Stack` untuk membuat, memperbarui, dan menghapus operasi
- `AWS::ElasticSearch::Domain` untuk memperbarui operasi
- `AWS::RDS::DBCluster` untuk membuat dan memperbarui operasi
- `AWS::RDS::DBInstance` untuk membuat, memperbarui, dan menghapus operasi
- `AWS::Redshift::Cluster` untuk memperbarui operasi

Operasi untuk sumber daya ini mungkin memerlukan waktu lebih lama daripada periode waktu tunggu default. Periode waktu tunggu bergantung pada sumber daya dan kredensial yang Anda gunakan. Untuk memperpanjang periode waktu habis, tentukan [peran layanan](#) saat Anda melakukan operasi tumpukan. Jika Anda sudah menggunakan peran layanan, atau jika tumpukan Anda berisi sumber daya yang tidak terdaftar, hubungi [AWS Dukungan](#).

Jika tumpukan Anda berada di status `UPDATE_ROLLBACK_FAILED`, lihat [Pembaruan Rollback Gagal](#).

Grup keamanan tidak ada di VPC

Verifikasi bahwa grup keamanan ada di VPC yang Anda tentukan. Jika grup keamanan ada, pastikan bahwa Anda menentukan ID grup keamanan dan bukan nama grup keamanan. Misalnya, sumber daya `AWS::EC2::SecurityGroupIngress` memiliki properti `SourceSecurityGroupName` dan `SourceSecurityGroupId`. Untuk grup keamanan VPC, Anda harus menggunakan properti `SourceSecurityGroupId` dan menentukan ID grup keamanan.

Pembaruan rollback gagal

Sumber daya dependen tidak dapat kembali ke status semula, menyebabkan rollback gagal (status `UPDATE_ROLLBACK_FAILED`). Misalnya, Anda mungkin memiliki tumpukan yang memutar kembali ke instance database lama yang dihapus di luar CloudFormation. Karena CloudFormation tidak tahu database telah dihapus, itu mengasumsikan bahwa instance database masih ada dan mencoba untuk memutar kembali ke sana, menyebabkan pembaruan rollback gagal.

Bergantung pada penyebab kegagalan, Anda dapat memperbaiki kesalahan secara manual dan melanjutkan rollback. Dengan melanjutkan rollback, Anda dapat mengembalikan tumpukan Anda ke status kerja (status `UPDATE_ROLLBACK_COMPLETE`), dan kemudian mencoba untuk memperbarui tumpukan lagi. Daftar berikut menjelaskan solusi untuk kesalahan umum yang menyebabkan kegagalan pembaruan rollback:

- Gagal menerima jumlah sinyal yang dibutuhkan

Gunakan perintah [sumber daya sinyal](#) untuk mengirim jumlah sinyal sukses yang diperlukan secara manual ke sumber daya yang menunggu mereka, dan kemudian lanjutkan memutar kembali pembaruan. Misalnya, selama rollback pembaruan, instance dalam grup Auto Scaling mungkin gagal memberi sinyal keberhasilan dalam durasi waktu tunggu yang ditentukan. Kirim sinyal sukses secara manual ke grup Auto Scaling. Ketika Anda melanjutkan rollback pembaruan, CloudFormation melihat sinyal Anda dan melanjutkan dengan rollback.

- Perubahan pada sumber daya dilakukan di luar CloudFormation

Sinkronkan sumber daya secara manual sehingga sesuai dengan templat tumpukan asli, lalu lanjutkan rollback pembaruan. Misalnya, jika Anda secara manual menghapus sumber daya yang CloudFormation mencoba memutar kembali, Anda harus secara manual membuat sumber daya tersebut dengan nama dan properti yang sama dengan yang dimilikinya di tumpukan asli.

- Izin tidak cukup

Periksa apakah Anda memiliki izin IAM yang cukup untuk memodifikasi sumber daya, dan kemudian lanjutkan pembaruan rollback. Misalnya, kebijakan IAM Anda mungkin memungkinkan Anda membuat bucket S3, tetapi tidak mengubah bucket. Tambahkan tindakan modifikasi ke kebijakan Anda.

- Token keamanan tidak valid

CloudFormation membutuhkan satu set kredensial baru. Tidak ada perubahan yang diperlukan. Lanjutkan rollback pembaruan, yang menyegarkan kredensial.

- Kesalahan pembatasan

Hapus sumber daya yang tidak Anda perlukan atau minta [peningkatan kuota](#), lalu lanjutkan memutar kembali pembaruan. Misalnya, jika kuota akun Anda untuk jumlah instans EC2 On-Demand adalah 5 dan rollback pembaruan melebihi kuota tersebut, maka akan gagal.

- Sumber daya tidak stabil

Sumber daya tidak merespons karena operasi mungkin telah melebihi periode CloudFormation batas waktu atau AWS layanan mungkin telah terganggu. Tidak ada perubahan yang diperlukan. Setelah operasi sumber daya selesai atau AWS layanan kembali beroperasi, lanjutkan memutar kembali pembaruan.

Untuk melanjutkan memutar kembali pembaruan, Anda dapat menggunakan CloudFormation konsol atau antarmuka baris AWS perintah (AWS CLI). Untuk informasi selengkapnya, lihat [Terus mengembalikan pembaruan](#).

Jika tidak ada solusi ini yang berhasil, Anda dapat melewati sumber daya yang CloudFormation tidak berhasil diputar kembali. Untuk informasi selengkapnya, lihat `ResourcesToSkip` parameter untuk operasi [ContinueUpdateRollback](#) API di Referensi AWS CloudFormation API. CloudFormation menetapkan status sumber daya yang ditentukan ke `UPDATE_COMPLETE` dan terus memutar kembali tumpukan. Setelah rollback selesai, status sumber daya yang dilewati akan tidak konsisten dengan status sumber daya di templat tumpukan. Sebelum Anda melakukan pembaruan tumpukan lain, Anda harus mengubah sumber daya atau memperbarui tumpukan agar konsisten satu sama lain. Jika tidak, pembaruan tumpukan berikutnya mungkin akan gagal dan membuat tumpukan Anda tidak dapat dipulihkan.

Kondisi tunggu tidak menerima jumlah sinyal yang diperlukan dari EC2 instans Amazon

Untuk mengatasi situasi ini, coba yang berikut ini:

- Pastikan AMI yang Anda gunakan telah menginstal skrip CloudFormation helper. Jika AMI tidak menyertakan skrip pembantu, Anda juga dapat mengunduhnya ke instans Anda. Untuk informasi selengkapnya, lihat [referensi skrip CloudFormation pembantu di Panduan Referensi AWS CloudFormation](#) Template.
- Verifikasi bahwa perintah `cfn-signal` berhasil dijalankan di instans. Anda dapat melihat log, seperti `/var/log/cloud-init.log` atau `/var/log/cfn-init.log`, untuk membantu Anda

men-debug peluncuran instans. Anda dapat mengambil log dengan masuk ke instance Anda, tetapi Anda harus [menonaktifkan rollback pada kegagalan](#) atau CloudFormation menghapus instance setelah tumpukan Anda gagal dibuat. Anda juga dapat [mempublikasikan log](#) ke Amazon CloudWatch. Untuk Windows, Anda dapat melihat log cfn C:\cfn\log dan log layanan EC2 Config. %ProgramFiles%\Amazon\EC2ConfigService

- Verifikasi bahwa instans tidak memiliki koneksi ke internet. Jika instans berada di VPC, instans harus dapat terhubung ke Internet melalui perangkat NAT jika berada di subnet pribadi atau melalui gateway Internet jika berada di subnet publik. Untuk menguji koneksi Internet instans, coba akses halaman web publik, seperti <http://aws.amazon.com>. Misalnya, Anda dapat menjalankan perintah berikut pada instans. Ini harus mengembalikan kode status HTTP 200.

```
curl -I https://aws.amazon.com
```

Untuk informasi tentang mengonfigurasi perangkat NAT, lihat [NAT di Panduan Pengguna](#) Amazon VPC.

Sumber daya dihilangkan dari tumpukan tetapi tidak dihapus

Selama pembaruan tumpukan, CloudFormation telah menghapus sumber daya dari tumpukan tetapi tidak menghapus sumber daya. Sumber daya lama masih ada, tetapi tidak lagi dapat diakses melalui CloudFormation. Hal ini dapat terjadi selama pembaruan tumpukan di mana:

- CloudFormation perlu mengganti sumber daya yang ada, jadi pertama-tama menciptakan sumber daya baru, kemudian mencoba menghapus sumber daya lama.
- Anda telah menghapus sumber daya dari templat tumpukan, jadi CloudFormation mencoba untuk menghapus sumber daya dari tumpukan.

Namun, mungkin ada kasus di mana tidak CloudFormation dapat menghapus sumber daya. Misalnya, jika pengguna tidak memiliki izin untuk menghapus sumber daya dari jenis tertentu.

CloudFormation mencoba untuk menghapus sumber daya lama tiga kali. Jika tidak CloudFormation dapat menghapus sumber daya lama, itu menghapus sumber daya lama dari tumpukan dan terus memperbarui tumpukan. Ketika pembaruan tumpukan selesai, CloudFormation mengeluarkan peristiwa UPDATE_COMPLETE tumpukan, tetapi menyertakan StatusReason yang menyatakan bahwa satu atau beberapa sumber daya tidak dapat dihapus. CloudFormation juga mengeluarkan

DELETE_FAILED acara untuk sumber daya tertentu, dengan yang sesuai StatusReason memberikan detail lebih lanjut tentang mengapa CloudFormation gagal menghapus sumber daya.

Untuk mengatasi situasi ini, hapus sumber daya langsung menggunakan konsol atau API untuk layanan yang mendasari.

Menghubungi dukungan

Jika sudah AWS Dukungan, Anda dapat membuat kasus dukungan teknis di [#/https://console.aws.amazon.com/support/rumah #/](https://console.aws.amazon.com/support/rumah). Sebelum Anda menghubungi dukungan, kumpulkan informasi berikut ini:

- ID tumpukan. Anda dapat menemukan ID tumpukan di tab Gambaran Umum dari [konsol CloudFormation](#). Untuk informasi selengkapnya, lihat [Lihat informasi tumpukan dari CloudFormation konsol](#).

Important

Jangan membuat perubahan pada tumpukan di luar CloudFormation. Membuat perubahan pada tumpukan Anda di luar CloudFormation mungkin menempatkan tumpukan Anda dalam keadaan tidak dapat dipulihkan.

- Pesan kesalahan tumpukan apa pun. Untuk informasi tentang melihat pesan kesalahan tumpukan, lihat bagian [Panduan pemecahan masalah](#).
- Untuk EC2 masalah Amazon, kumpulkan log cloud-init dan cfn. Log ini dipublikasikan di EC2 instance Amazon di /var/log/ direktori. Log ini menangkap proses dan output perintah saat instans Anda sedang disiapkan. Untuk Windows, kumpulkan layanan EC2 Konfigurasi dan log cfn %ProgramFiles%\Amazon\EC2ConfigService dan C:\cfn\log.

Anda juga dapat mencari jawaban dan memposting pertanyaan di [CloudFormation komunitas](#) di AWS re:Post.

Memecahkan masalah penerapan CloudFormation tumpukan yang gagal dengan Pengembang Amazon Q

Amazon Q Developer adalah asisten percakapan yang didukung kecerdasan buatan (AI) generatif yang dapat membantu Anda memahami, membangun, memperluas, dan mengoperasikan AWS aplikasi. Untuk informasi tentang Pengembang Amazon Q, lihat [Panduan Pengguna Pengembang Amazon Q](#).

Dokumen ini menjelaskan cara menggunakan Amazon Q Developer untuk membantu Anda memecahkan masalah penerapan yang gagal CloudFormation . Fitur ini dirancang untuk membantu Anda mengidentifikasi dan menyelesaikan masalah yang dihadapi dengan cepat selama penerapan tumpukan.

Didukung oleh Amazon Bedrock: Amazon Q dibangun di Amazon Bedrock dan mencakup [deteksi penyalahgunaan otomatis](#) yang diterapkan di Amazon Bedrock untuk menegakkan keselamatan, keamanan, dan penggunaan AI yang bertanggung jawab.

Topik

- [Fitur](#)
- [Cara kerjanya](#)
- [Harga](#)
- [Ketersediaan wilayah](#)
- [Batasan](#)

Fitur

Penyebaran tumpukan pemecahan masalah dengan solusi Pengembang Amazon Q menyediakan fitur berikut:

- Analisis kesalahan - Menyediakan easy-to-understand analisis kesalahan penerapan CloudFormation tumpukan.
- Saran resolusi — Menawarkan step-by-step instruksi untuk menyelesaikan masalah yang teridentifikasi.

Cara kerjanya

Ketika pembuatan CloudFormation tumpukan atau pembaruan gagal, pesan kesalahan ditampilkan di file AWS Management Console.

Anda dapat memilih tombol Diagnosa dengan Q di sebelah pesan kesalahan.

Amazon Q Developer menganalisis kesalahan dan memberikan penjelasan tentang masalah ini.

Untuk bantuan tambahan, pilih Bantu saya menyelesaikan untuk menerima langkah-langkah resolusi terperinci.

Harga

Untuk informasi tentang harga, lihat [harga Pengembang Amazon Q](#).

Ketersediaan wilayah

Pengembang Amazon Q tersedia di sebagian besar Wilayah AWS. Untuk daftar semua Wilayah tempat Pengembang Amazon Q saat ini tersedia, lihat [Wilayah yang Didukung untuk Pengembang Amazon Q](#) di Panduan Pengguna Pengembang Amazon Q.

Batasan

- Fitur ini hanya dapat diakses melalui AWS Management Console.
- Fitur ini mencakup 20 kesalahan paling umum teratas. Itu tidak mencakup semua kemungkinan kesalahan.
- Kasus pemecahan masalah yang kompleks mungkin masih memerlukan Dukungan intervensi.

Riwayat dokumen untuk Panduan AWS CloudFormation Pengguna

Tabel berikut menjelaskan perubahan penting pada konten Panduan AWS CloudFormation Pengguna setelah Mei 2018. Untuk menerima pemberitahuan tentang pembaruan dokumentasi, Anda dapat berlangganan umpan RSS.

Important

Baris tabel yang menjelaskan pembaruan pada konten referensi template mulai Mei 2018 dan seterusnya telah pindah ke Referensi AWS CloudFormation Template yang baru. Untuk pembaruan ini dan perubahan apa pun di masa mendatang, lihat [Riwayat dokumen untuk Referensi AWS CloudFormation Template di Referensi AWS CloudFormation Template](#).

Perubahan	Deskripsi	Tanggal
Panduan PDF tersedia	Anda sekarang dapat mengunduh Panduan AWS CloudFormation Pengguna sebagai PDF.	30 Mei 2025
Memindahkan konten referensi template ke panduan baru	CloudFormation menerbitkan Panduan Referensi AWS CloudFormation Template. Untuk detailnya, lihat Panduan Referensi AWS CloudFormation Template .	30 Mei 2025
Generator IAc mendukung pemindaian sebagian	Anda sekarang dapat memilih jenis sumber daya tertentu untuk dipindai, sehingga lebih mudah untuk menghasilkan templat Infrastructure-as-Code (IAc) dari sumber daya yang ada. Untuk informasi	Maret 27, 2025

selengkapnya, lihat [Memulai pemindaian sumber daya dengan generator CloudFormation IAc](#).

[Stack refactoring](#)

Stack refactoring menyederhanakan penataan ulang sumber daya di CloudFormation tumpukan Anda sambil tetap mempertahankan properti dan data sumber daya yang ada. Untuk informasi selengkapnya, lihat [Stack refactoring](#).

Februari 6, 2025

[Memecahkan masalah penerapan tumpukan dengan Pengembang Amazon Q](#)

Anda sekarang dapat menggunakan Pengembang Amazon Q untuk memecahkan masalah kesalahan umum saat menerapkan CloudFormation tumpukan. Untuk informasi selengkapnya, lihat [Memecahkan masalah penerapan CloudFormation tumpukan yang gagal dengan Amazon Q Developer](#).

November 22, 2024

[Grafik garis waktu penyebaran tumpukan](#)

Anda sekarang dapat melihat representasi visual dari penyebaran tumpukan Anda. Grafik timeline penerapan tumpukan menunjukkan status penerapan tumpukan, status penerapan sumber daya individual, dan waktu status penerapan berubah. Untuk informasi selengkapnya, [lihat Melihat garis waktu penerapan CloudFormation tumpukan](#).

November 11, 2024

[Visualisasikan sumber daya yang dipindai dan templat yang dihasilkan](#)

Sekarang Anda dapat merampingkan alur kerja generator Infrastructure as Code (IaC) dengan memvisualisasikan detail ringkasan pemindaian dan melihat pratinjau templat yang dihasilkan sebelum menerapkan tumpukan infrastruktur Anda. Untuk informasi selengkapnya, lihat [Melihat ringkasan pemindaian di CloudFormation konsol](#) dan [Membuat CloudFormation tumpukan dari sumber daya yang dipindai](#).

Agustus 22, 2024

[EventBridge Integrasi
Amazon dengan sinkronisasi
AWS CloudFormation Git](#)

AWS CloudFormation Git sync sekarang menerbitkan perubahan status sinkronisasi sebagai peristiwa ke Amazon EventBridge. Untuk informasi selengkapnya, lihat [detail peristiwa Perubahan Status Sinkronisasi Repositori dan detail peristiwa Perubahan Status Sinkronisasi Sumber Daya](#).

Juli 29, 2024

[Paksa hapus tumpukan yang macet](#)

Dua opsi baru untuk memaksa menghapus tumpukan tersedia untuk operasi penghapusan tumpukan yang macet. Anda sekarang dapat memilih untuk memaksa menghapus tumpukan tetapi mempertahankan sumber daya, atau menghapus kekuatan seluruh tumpukan. Untuk informasi selengkapnya, lihat [Menghapus tumpukan dari CloudFormation konsol](#).

22 Mei 2024

[AWS CloudTrail penyebab akar operasi tumpukan peristiwa](#)

CloudFormation meningkatkan pengalaman pemecahan masalah untuk operasi tumpukan dengan integrasi AWS CloudTrail deep-link baru. Fitur ini secara langsung menautkan peristiwa operasi tumpukan di CloudFormation konsol ke CloudTrail peristiwa yang relevan. Untuk informasi selengkapnya, lihat [Menentukan penyebab kegagalan tumpukan](#).

15 Mei 2024

[Set perubahan tingkat properti](#)

Set perubahan tingkat properti memungkinkan Anda untuk melihat pratinjau perubahan yang akan CloudFormation dilakukan penerapan pada nilai properti sumber daya. Untuk informasi selengkapnya, lihat [Melihat set perubahan untuk CloudFormation tumpukan](#).

April 12, 2024

[CloudFormation memperkenalkan acara CONFIGURATION_COMPLETE](#)

Anda sekarang dapat menggunakan CONFIGURATION_COMPLETE acara untuk mengaktifkan alur kerja yang lebih cepat yang melibatkan pembuatan sumber daya. Untuk informasi selengkapnya, lihat [Memahami peristiwa pembuatan CloudFormation tumpukan](#).

Maret 11, 2024

[Hasilkan AWS CloudFormation templat dan AWS CDK aplikasi dari AWS sumber daya yang ada](#)

Anda sekarang dapat membuat template menggunakan sumber daya yang disediakan di akun Anda yang belum dikelola oleh CloudFormation. Untuk informasi selengkapnya, lihat [Menghasilkan templat dari sumber daya yang ada dengan generator IAc](#).

Februari 2, 2024

[StackSets modus konkurensi](#)

Mode Konkurensi adalah parameter `StackSetOperationPreferences` yang memungkinkan Anda memilih bagaimana tingkat konkurensi berperilaku selama operasi set tumpukan. Untuk informasi selengkapnya, lihat [Memilih Mode Konkurensi untuk CloudFormation StackSets](#).

9 November 2023

[Informasi StackSet drift terperinci](#)

Berikut ini APIs memungkinkan Anda untuk melihat instance tumpukan mana yang telah hanyut dari StackSet template dan sumber daya mana yang telah hanyut.

Juli 24, 2023

[ListStackInstanceResourceDrifts](#)

Mengembalikan informasi drift untuk sumber daya dalam instance stack.

[StackInstanceResourceDrifts Summary](#)

Struktur yang berisi informasi ringkasan tentang drift sumber daya untuk instance tumpukan.

[CloudFormation StackSets](#)
[APIs untuk mengontrol akses](#)
[AWS Organizations kepercayaan](#)
[an](#)

CloudFormation StackSets menyediakan pelanggan dengan hal-hal berikut APIs untuk mengelola akses AWS Organizations kepercayaan:

Juni 5, 2023

[ActivateOrganizationsAccess](#)

Aktifkan akses tepercaya dengan AWS Organizations. Dengan akses tepercaya antara StackSets dan Organizations diaktifkan, akun manajemen memiliki izin untuk membuat dan mengelola StackSets organisasi Anda.

[DeactivateOrganizationsAccess](#)

Menonaktifkan akses tepercaya dengan AWS Organizations. Jika akses tepercaya dinonaktifkan, akun manajemen tidak memiliki izin untuk membuat dan mengelola layanan yang dikelola StackSets untuk organisasi Anda.

[DescribeOrganizationsAccess](#)

Mengambil informasi tentang OrganizationAccess status akun. API ini dapat

dipanggil baik oleh akun manajemen atau administrator yang didelegasikan dengan menggunakan `CallAs` parameter. API ini juga dapat dipanggil tanpa `CallAs` parameter oleh akun manajemen.

[DescribeStackSet API](#)

`DescribeStackSet` API memiliki parameter baru ke daftar Wilayah tempat kumpulan tumpukan tertentu diterapkan. Untuk informasi selengkapnya, lihat [DescribeStackSet](#).

1 Februari 2023

[Mengelola StackSets acara dengan CloudFormation dan Amazon EventBridge](#)

CloudFormation StackSets meluncurkan pemberitahuan acara melalui Amazon EventBridge. Anda dapat memicu tindakan berbasis peristiwa setelah membuat, memperbarui, atau menghapus kumpulan tumpukan Anda CloudFormation. Untuk informasi selengkapnya, lihat [Monitoring CloudFormation dan peristiwa sinkronisasi Git dengan EventBridge](#).

16 November 2022

[Peningkatan wawasan tentang instance tumpukan untuk operasi set tumpukan](#)

CloudFormation StackSets memberikan informasi lebih rinci tentang instance tumpukan untuk operasi set tumpukan:

4 November 2022

[DescribeStackSetOperation](#)

Anda sekarang dapat menggunakan `DescribeStackSetOperation` untuk memberikan jumlah instance tumpukan yang gagal untuk operasi set tumpukan selama penerapan.

[ListStackInstances](#)

Anda sekarang dapat menggunakan opsi penyaringan `LastOperationID` untuk membuat daftar instance tumpukan untuk operasi set tumpukan.

[Mengelola acara dengan CloudFormation dan Amazon EventBridge](#)

Terima pemberitahuan saat CloudFormation peristiwa tertentu terjadi. Untuk informasi selengkapnya, lihat [Monitoring CloudFormation dan peristiwa sinkronisasi Git dengan EventBridge](#).

20 Juli 2022

[Tingkat akun](#)

CloudFormation mengumumkan ketersediaan umum jenis filter akun, fitur yang memungkinkan pelanggan membatasi target penyebaran ke akun individu atau menyertakan akun tambahan yang disediakan. OUs

Untuk informasi selengkapnya, lihat [Target level akun untuk layanan yang dikelola StackSets](#).

Juli 7, 2022

[CloudFormation registri](#)

CloudFormation mengumumkan ketersediaan umum Hooks, fitur yang memungkinkan pelanggan untuk memanggil logika khusus untuk mengotomatiskan tindakan atau memeriksa konfigurasi sumber daya sebelum membuat, memperbarui, atau menghapus operasi tumpukan.

Untuk informasi selengkapnya, lihat [Panduan Pengguna AWS CloudFormation Hooks](#).

Februari 10, 2022

[Opsi kegagalan tumpukan](#)

Anda dapat mengembankan aplikasi secara iteratif saat kegagalan penyediaa n ditemui dengan memulai dari titik kegagalan tanpa mengembalikan sumber daya yang berhasil disediakan. Dengan menentukan opsi kegagalan tumpukan, Anda dapat memecahkan masalah sumber daya dalam status `CREATE_FAILED` atau `UPDATE_FAILED`. Anda dapat menyediakan opsi kegagalan untuk semua penerapan tumpukan dan mengubah operasi set. Untuk informasi selengkapnya, lihat [Memilih cara menangani kegagalan saat menyediakan sumber daya](#).

Agustus 30, 2021

[Impor tumpukan ke set tumpukan](#)

Anda sekarang dapat mengimpor tumpukan yang ada ke set tumpukan baru atau yang sudah ada. Untuk informasi selengkapnya, lihat [Mengimpor tumpukan ke](#) dalam `CloudFormation StackSets`

28 Juli 2021

Peningkatan kuota	Anda sekarang dapat mendeklarasikan maksimum tumpukan default di Anda. 2000 Akun AWS Untuk informasi selengkapnya, lihat Memahami CloudFormation kuota .	15 Juli 2021
Publikasikan ekstensi pihak ketiga publik	Anda sekarang dapat menggunakan ekstensi publik yang disediakan oleh penerbit pihak ketiga, seperti halnya ekstensi Anda. AWS Untuk informasi selengkapnya, lihat Menggunakan ekstensi publik pihak ketiga dari CloudFormation registri .	21 Juni, 2021
Makro referensi dalam templat set tumpukan	StackSets sekarang mendukung pembuatan atau pembaruan set tumpukan dengan izin yang dikelola sendiri dari template yang mereferensikan makro. Untuk informasi selengkapnya tentang makro, lihat Melakukan pemrosesan kustom pada CloudFormation templat dengan makro templat .	14 April 2021

[Menggunakan nilai terbaru dari parameter Systems Manager dalam referensi dinamis](#)

Anda sekarang dapat CloudFormation menggunakan an versi terbaru dari parameter Systems Manager setiap kali Anda membuat atau memperbarui tumpukan. Anda tidak lagi diharuskan untuk menentukan versi tertentu. Untuk detail selengkapnya, lihat [Mendapatkan nilai teks biasa dari Systems Manager Parameter Store](#).

13 April, 2021

[Dukungan modul menggunakan pembatas periode dalam nama sumber daya](#)

Anda sekarang dapat menggunakan periode sebagai pembatas dalam menentukan nama logis yang sepenuhnya memenuhi syarat untuk sumber daya yang terkandung dalam modul. Untuk informasi selengkapnya, lihat [Sumber daya modul referensi di CloudFormation templat](#).

8 April 2021

[CloudFormation StackSets sekarang mendukung penyebaran wilayah paralel](#)

Anda sekarang dapat memilih untuk menyebarkan StackSets ke Wilayah secara berurutan atau paralel. Untuk informasi selengkapnya, lihat [Ops operasi set tumpukan](#).

6 April 2021

[CloudFormation StackSets
sekarang mendukung
administrator yang didelegas
ikan dengan AWS Organizat
ions](#)

Selain akun manajemen organisasi, akun administrator yang didelegasikan dapat membuat dan mengelola kumpulan tumpukan dengan izin yang dikelola layanan untuk organisasi mereka. Untuk selengkapnya, lihat [Mendaftarkan akun anggota administrator yang didelegasikan](#) dan [Membuat CloudFormation StackSets dengan izin yang dikelola layanan](#).

18 Februari 2021

[CloudFormation StackSets
Ketersediaan Wilayah](#)

CloudFormation StackSets sekarang tersedia di Wilayah Asia Pasifik (Osaka). Untuk informasi selengkapnya, lihat [Mengelola tumpukan di seluruh akun dan Wilayah dengan StackSets](#).

10 Februari 2021

Modul

24 November 2020

Modul adalah cara bagi Anda untuk mengemas konfigurasi sumber daya untuk dimasukkan ke seluruh templat tumpukan, dengan cara yang transparan, mudah dikelola, dan dapat diulang. Modul dapat merangkum konfigurasi layanan umum dan praktik terbaik sebagai blok bangunan modular yang dapat disesuaikan untuk Anda masukkan dalam templat tumpukan Anda. Untuk informasi selengkapnya, lihat [Membuat konfigurasi sumber daya yang dapat digunakan kembali yang dapat disertakan di seluruh templat dengan CloudFormation modul](#).

Ubah set untuk tumpukan bersarang

18 November 2020

Dengan set perubahan untuk tumpukan nest Anda dapat meninjau perubahan aplikasi dan sumber daya infrastruktur Anda di seluruh hirarki tumpukan nest dan melanjutkan dengan pembaruan ketika Anda telah mengonfirmasi bahwa semua perubahan sesuai dengan yang ditujukan. Untuk informasi selengkapnya, lihat [Set perubahan untuk tumpukan nest](#).

Kuota meningkat

AWS CloudFormation Kuota berikut telah diperbarui.

22 Oktober 2020

- Anda sekarang dapat menyatakan maksimal pemetaan 200 di templat AWS CloudFormation Anda.
- Anda sekarang dapat mendeklarasikan maksimum atribut 200 pemetaan untuk setiap pemetaan dalam template Anda. AWS CloudFormation
- Anda sekarang dapat menyatakan maksimal output 200 di templat AWS CloudFormation Anda.
- Anda sekarang dapat menyatakan maksimal parameter 200 di templat AWS CloudFormation Anda.
- Anda sekarang dapat menyatakan maksimal sumber daya 500 di templat AWS CloudFormation Anda.
- Anda sekarang dapat melewati badan template dengan ukuran maksimum 1 MB dalam objek Amazon S3.

[Deteksi drift untuk sumber daya pribadi](#)

CloudFormation sekarang mendukung operasi deteksi drift pada daftar AWS sumber daya yang diperluas, serta sumber daya pribadi yang didefinisikan sebagai dapat ditentukan dalam registri. CloudFormation Untuk informasi selengkapnya, lihat [Dukungan jenis sumber daya](#).

1 Oktober 2020

[Izin yang diperbarui diperlukan untuk mendaftarkan penyedia sumber daya](#)

Mendaftarkan penyedia sumber daya di akun Anda sekarang mengharuskan Anda memiliki izin untuk mengakses paket handler skema yang diunggah ke bucket S3 untuk penyedia sumber daya tersebut. Untuk informasi selengkapnya, lihat [Izin IAM untuk mendaftarkan ekstensi pribadi pihak ketiga](#).

7 Agustus 2020

[Impor sumber daya mendukung jenis sumber daya pribadi yang dapat disediakan](#)

Operasi impor sekarang mendukung tipe sumber daya privat yang tersedia; yaitu, yang tipe penyediaannya adalah FULLY_MUTABLE atau IMMUTABLE . Untuk informasi selengkapnya, lihat [Sumber daya yang mendukung operasi impor](#).

3 Juni 2020

[blue/green Penerapan ECS melalui CodeDeploy](#)

Anda sekarang dapat menggunakan CloudFormation untuk melakukan blue/green penyebaran ECS melalui CodeDeploy. Blue/green penerapan adalah strategi penyebaran aman yang disediakan oleh AWS CodeDeploy untuk meminimalkan gangguan yang disebabkan oleh perubahan versi aplikasi. Untuk informasi selengkapnya, lihat [Melakukan blue/green penerapan ECS melalui penggunaan CodeDeploy CloudFormation](#).

19 Mei 2020

[CloudFormation StackSets Ketersediaan Wilayah](#)

CloudFormation StackSets sekarang tersedia di Wilayah AWS GovCloud (AS-Barat).

18 Mei 2020

[AWS CloudFormation
StackSets Integrasi dengan
AWS Organizations](#)

Sekarang Anda dapat menggunakan StackSets untuk mengelola penerapan secara terpusat ke semua akun di organisasi atau unit organisasi tertentu (OUs) di AWS Organizations. Anda dapat mengaktifkan penerapan otomatis ke akun baru yang ditambahkan ke organisasi Anda atau OUs Izin yang diperlukan untuk menyebarkan di seluruh akun akan ditangani secara otomatis oleh StackSets. Untuk informasi selengkapnya, lihat [Mengelola tumpukan di seluruh akun dan Wilayah dengan StackSets](#).

11 Februari 2020

[Deteksi Drift untuk StackSets](#)

Anda sekarang dapat menjalankan deteksi drift pada set tumpukan dan semua instans tumpukan yang dimasukkan. Untuk informasi selengkapnya, lihat [Melakukan deteksi drift pada CloudFormation StackSets](#).

19 November 2019

[CloudFormation registri
sekarang tersedia](#)

Anda sekarang dapat menggunakan CloudFormation konsol untuk melihat sumber daya pribadi dan publik yang tersedia untuk digunakan di akun Anda. Untuk informasi selengkapnya, [lihat Melihat ekstensi yang tersedia dan diaktifkan di CloudFormation registri](#).

18 November 2019

[CloudFormation tindakan API registri](#)

Tindakan API berikut untuk mengelola tipe dalam CloudFormation registri sekarang tersedia.

18 November 2019

[DeregisterType](#)

Menghapus jenis atau jenis versi dari penggunaan aktif dalam CloudFormation registri.

[DescribeType](#)

Mengembalikan informasi rinci tentang tipe terdaftar.

[DescribeTypeRegistration](#)

Mengembalikan informasi tentang pendaftaran tipe, termasuk statusnya saat ini dan pengidentifikasi tipe dan versinya.

[ListTypeRegistrations](#)

Mengembalikan daftar pengidentifikasi permintaan pendaftaran untuk tipe tertentu.

[ListTypes](#)

Mengembalikan informasi ringkasan tentang jenis yang telah terdaftar CloudFormation.

[ListTypeVersions](#)

Mengembalikan informasi ringkasan tentang versi tipe.

[RegisterType](#)

Mendaftarkan tipe dengan CloudFormation registri. Mendaftarkan jenis membuatnya tersedia untuk digunakan dalam CloudFormation template di Anda Akun AWS.

[SetTypeDefaultVersion](#)

Tentukan versi tipe default. Versi default dari suatu tipe akan digunakan dalam CloudFormation operasi.

Untuk informasi selengkap nya tentang CloudFormation registri, lihat [Mengelola ekstensi dengan CloudFormation registri](#)

[Impor sumber daya ditambahkan](#)

Jika Anda membuat AWS sumber daya di luar CloudFormation manajemen, Anda dapat membawa sumber daya yang ada ini ke dalam CloudFormation manajemen menggunakan `anresource import`. Untuk informasi selengkapnya, lihat [Mengimpor AWS sumber daya ke CloudFormation tumpukan dengan impor sumber daya](#).

11 November 2019

[Batas set tumpukan meningkat](#)

Anda sekarang dapat membuat maksimum 100 set tumpukan di akun administrator Anda, membuat maksimum 2000 instans tumpukan per set tumpukan, dan menjalankan maksimum 3500 operasi instans tumpukan di setiap wilayah pada waktu yang sama, per akun administrator. Untuk informasi selengkapnya, lihat [Memahami CloudFormation kuota](#).

Selasa, 02 Agustus 2019

[Batasi sumber daya dalam operasi tumpukan bersamaan](#)

CloudFormation sekarang memberlakukan batas akun untuk jumlah sumber daya dalam operasi tumpukan bersamaan. Batas ini ditentukan oleh wilayah. Untuk informasi selengkapnya, lihat [Memahami CloudFormation kuota](#).

30 April 2019

[Batas operasi instance stack](#)

Untuk StackSets, Anda sekarang dapat memiliki maksimum 1500 operasi instans tumpukan yang berjalan di wilayah tertentu pada saat yang sama, per akun administrator. Untuk informasi selengkapnya, lihat [Memahami CloudFormation kuota](#).

13 Desember 2018

[CAPABILITY_AUTO_EXPAND Kemampuan sekarang tersedia](#)

Anda sekarang dapat menggunakan CAPABILITY_AUTO_EXPAND kemampuan untuk membuat atau memperbarui tumpukan langsung dari template tumpukan yang berisi makro, tanpa terlebih dahulu meninjau perubahan yang dihasilkan dalam set perubahan terlebih dahulu. Untuk informasi selengkapnya, lihat [CreateStack](#) atau [UpdateStack](#) di Referensi AWS CloudFormation API.

7 Desember 2018

[Deteksi drift tumpukan ditambahkan](#)

13 November 2018

Anda sekarang dapat mendeteksi apakah konfigurasi sebenarnya tumpukan telah melayang dari konfigurasi template yang diharapkan seperti yang didefinisikan di dalamnya. CloudFormation Anda dapat mendeteksi penyimpangan pada seluruh tumpukan, atau sumber daya tumpukan individu. Untuk informasi selengkapnya, lihat [Mendeteksi perubahan konfigurasi yang tidak dikelola pada tumpukan dan sumber daya dengan deteksi drift.](#)

[secretsmanager referensi dinamis sekarang tersedia](#)

Anda sekarang dapat menggunakan referensi `secretsmanager` dinamis untuk mengambil seluruh rahasia atau nilai rahasia yang disimpan di AWS Secrets Manager. Rahasia dapat berupa kredensial basis data, kata sandi, kunci API pihak ketiga, dan bahkan teks acak. Menggunakan referensi `secretsmanager` dinamis menjamin bahwa Secrets Manager atau CloudFormation mencatat atau mempertahankan nilai rahasia yang diselesaikan. Untuk informasi selengkapnya, lihat [Mendapatkan nilai rahasia atau rahasia dari Secrets Manager](#).

9 November 2018

[Makro sekarang tersedia](#)

Anda sekarang dapat menggunakan makro untuk melakukan pemrosesan kustom pada templat, dari tindakan sederhana seperti find-and-replace operasi hingga transformasi ekstensif seluruh templat. Untuk informasi selengkapnya, lihat [Melakukan pemrosesan kustom pada CloudFormation templat dengan makro templat](#).

6 September 2018

[CloudFormation sekarang mendukung titik akhir VPC yang didukung oleh PrivateLink](#)

Anda dapat menggunakan VPC endpoint untuk membuat koneksi privat antara VPC dan CloudFormation Anda tanpa memerlukan akses melalui Internet, melalui instans NAT, koneksi VPN, atau AWS Direct Connect. Untuk informasi selengkapnya, lihat [Akses CloudFormation menggunakan titik akhir antarmuka \(AWS PrivateLink\)](#).

22 Agustus 2018

[Referensi dinamis mendukung string aman](#)

Sekarang Anda dapat menggunakan referensi dinamis baru untuk menentukan nilai yang disimpan dan dikelola di layanan lain, termasuk parameter SecureString tipe Systems Manager Parameter Store, di template tumpukan Anda. Untuk informasi selengkapnya, lihat [Mendapatkan nilai string aman dari Systems Manager Parameter Store](#).

16 Agustus 2018

[Stack set sekarang mendukung peran eksekusi yang disesuaikan](#)

Sekarang Anda dapat menggunakan peran eksekusi yang disesuaikan di akun target untuk mengontrol sumber daya tumpukan yang dapat disertakan pengguna atau grup dalam kumpulan tumpukan mereka. Untuk informasi selengkapnya, lihat [Menyiapkan opsi izin lanjutan untuk operasi set tumpukan](#).

30 Mei 2018

[Pembaruan selektif dari instance tumpukan](#)

Anda sekarang dapat menggunakan opsional Accounts dan Regions parameter untuk menentukan akun dan wilayah untuk memperbarui instance tumpukan selama operasi pembaruan set tumpukan. Untuk informasi selengkapnya, lihat [UpdateStackSet](#) di dalam Referensi API AWS CloudFormation .

30 Mei 2018

[CloudFormation sekarang membuat bucket S3 dengan enkripsi diaktifkan](#)

Untuk bucket Amazon S3 yang CloudFormation dibuat untuk menyimpan templat tumpukan yang diunggah, enkripsi sisi server sekarang diaktifkan secara default, sehingga mengenkripsi semua objek yang disimpan dalam bucket tersebut. Untuk informasi selengkapnya, lihat [Membuat tumpukan dari CloudFormation konsol](#).

24 Mei 2018

[Titik akhir FIPS ditambahkan](#)

CloudFormation sekarang menawarkan titik akhir baru yang menggunakan modul kriptografi tervalidasi FIPS 140-2 di wilayah publik AS berikut: US-East-1, US-East-2, US-West-1, dan US-Barat-2. Lihat [AWS CloudFormation titik akhir dan kuota](#) di titik akhir baru yang Referensi Umum Amazon Web Services sesuai dengan FIPS. URLs

17 Mei 2018

Untuk pembaruan Panduan Pengguna AWS CloudFormation Hooks, lihat Riwayat dokumen untuk Panduan [Pengguna AWS CloudFormation Hooks di Panduan Pengguna AWS CloudFormation Hooks](#).

Pembaruan yang diarsipkan

Tabel berikut menjelaskan perubahan penting dalam setiap rilis Panduan AWS CloudFormation Pengguna sebelum Mei 2018.

Perubahan	Tanggal rilis	Deskripsi	Versi API
Sumber daya yang diperbarui	22 Juli 2019	Gunakan <code>encryptionOptions</code> properti untuk menentukan Kunci milik AWS atau kunci yang dikelola pelanggan untuk broker Amazon MQ.	2010-05-15
Konvensi penamaan set tumpukan	10 April 2018	CloudFormation tumpukan yang dibuat menggunakan set tumpukan sekarang mengikuti konvensi penamaan baru, di mana nama tumpukan berisi nama set tumpukan.	2010-05-15
Sumber daya baru	10 April 2018	AWS::AppSync::ApiKey Gunakan <code>AWS::AppSync::ApiKey</code> sumber daya untuk membuat kunci unik yang dapat Anda distribusikan ke klien yang menjalankan operasi GraphQL. AWS AppSync AWS::AppSync::DataSource Gunakan <code>AWS::AppSync::DataSource</code> sumber daya untuk membuat sumber data untuk resolver di. AWS AppSync AWS::AppSync::GraphQLApi Gunakan <code>AWS::AppSync::GraphQLApi</code> sumber daya untuk membuat AWS AppSync GraphQL API baru. AWS::AppSync::GraphQLSchema Gunakan <code>AWS::AppSync::GraphQLSchema</code> sumber daya untuk membuat model data untuk AWS AppSync GraphQL API Anda.	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
		AWS::AppSync::Resolver Gunakan sumber daya <code>AWS::AppSync::Resolver</code> untuk menentukan resolver GraphQL logis yang akan Anda lampirkan ke bidang dalam skema.	
Sumber daya yang diperbarui	10 April 2018	AWS::Config::ConfigurationAggregator Gunakan tipe <code>OrganizationAggregationSource</code> properti untuk menentukan wilayah AWS Config data yang akan digabungkan menjadi agregator AWS Config konfigurasi dan peran IAM yang akan digunakan untuk mengambil detail. AWS Organizations	2010-05-15
Sumber daya baru	4 April 2018	AWS::Config::AggregationAuthorization Gunakan <code>AWS::Config::AggregationAuthorization</code> sumber daya untuk memberikan izin ke akun agregator untuk mengumpulkan AWS Config data Anda. AWS::Config::ConfigurationAggregator Gunakan sumber daya <code>AWS::Config::ConfigurationAggregator</code> untuk membuat agregator konfigurasi untuk AWS Config.	2010-05-15
Set tumpukan sekarang mendukung peran administrator yang disesuaikan	29 Maret 2018	Gunakan peran administrator yang disesuaikan untuk mengendalikan pengguna atau grup mana yang mengelola set tumpukan tertentu dalam akun administrator yang sama. Untuk informasi selengkapnya, lihat Menyiapkan opsi izin lanjutan untuk operasi set tumpukan .	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Sumber daya baru	29 Maret 2018	AWS::EC2::LaunchTemplate Gunakan <code>AWS::EC2::LaunchTemplate</code> sumber daya untuk membuat template peluncuran untuk EC2 instance Amazon.	2010-05-15
Sumber daya yang diperbarui	29 Maret 2018	AWS::AutoScaling::AutoScalingGroup Gunakan properti <code>LaunchTemplate</code> untuk menentukan templat peluncuran yang akan digunakan untuk meluncurkan instans. AWS::EC2::SpotFleet Dalam tipe SpotFleetRequestConfigData properti, gunakan <code>LaunchTemplateConfigs</code> properti untuk mendeskripsikan template peluncuran dan penggantian.	2010-05-15
Fungsi intrinsik <code>Fn::Cidr</code> baru	6 Maret 2018	Mengembalikan blok alamat Cidr yang ditentukan. Untuk informasi selengkapnya, lihat Fn::Cidr .	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Sumber daya baru	6 Maret 2018	AWS::ApiGateway::VpcLink Gunakan sumber daya <code>AWS::ApiGateway::VpcLink</code> untuk menentukan tautan VPC API Gateway bagi <code>AWS::ApiGateway::RestApi</code> untuk mengakses sumber daya di Amazon Virtual Private Cloud (VPC). AWS::GuardDuty::Master Gunakan <code>AWS::GuardDuty::Master</code> sumber daya untuk membuat akun GuardDuty utama. AWS::GuardDuty::Member Gunakan <code>AWS::GuardDuty::Member</code> sumber daya untuk membuat akun GuardDuty anggota. AWS::SES::ConfigurationSet Gunakan <code>AWS::SES::ConfigurationSet</code> sumber daya untuk membuat grup aturan yang dapat Anda terapkan pada email yang Anda kirim. AWS::SES::ConfigurationSetEventDestination Gunakan sumber daya <code>AWS::SES::ConfigurationSetEventDestination</code> untuk menentukan tujuan acara set konfigurasi. AWS::SES::ReceiptFilter Gunakan sumber daya <code>AWS::SES::ReceiptFilter</code> untuk menentukan apakah akan menerima atau menolak surat yang berasal dari alamat IP atau berbagai alamat IP.	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
		AWS::SES::ReceiptRule Gunakan <code>AWS::SES::ReceiptRule</code> sumber daya untuk menentukan tindakan yang harus dilakukan Amazon SES saat menerima email atas nama satu atau beberapa alamat email atau domain yang Anda miliki. AWS::SES::ReceiptRuleSet Gunakan <code>AWS::SES::ReceiptRuleSet</code> sumber daya untuk menentukan aturan kosong yang ditetapkan untuk Amazon SES. AWS::SES::Template Gunakan <code>AWS::SES::Template</code> sumber daya untuk menentukan konten email, terdiri dari baris subjek, bagian HTML, dan bagian teks saja.	

Perubahan	Tanggal rilis	Deskripsi	Versi API
Sumber daya yang diperbarui	6 Maret 2018	AWS::AutoScaling::AutoScalingGroup Gunakan properti <code>AutoScalingGroupName</code> untuk menentukan nama grup Auto Scaling. AWS::ApiGateway::RestApi Gunakan properti <code>ApiKeySourceType</code> untuk menentukan sumber kunci API untuk permintaan pengukuran sesuai dengan rencana penggunaan. Gunakan <code>MinimumCompressionSize</code> properti untuk menentukan bilangan bulat nullable yang digunakan untuk mengaktifkan kompresi atau menonaktifkan kompresi pada API. AWS::ApplicationAutoScaling::ScalingPolicy Dalam tipe TargetTrackingScalingPolicyConfiguration properti, gunakan <code>DisableScaleIn</code> properti untuk menentukan apakah skala dalam kebijakan pelacakan target dinonaktifkan. AWS::EC2::SpotFleet Dalam jenis LaunchSpecifications properti, gunakan <code>TagSpecifications</code> properti untuk menentukan tag yang akan diterapkan selama <code>SpotFleet</code> pembuatan. AWS::Elasticsearch::Domain Gunakan atribut <code>Arn</code> agar <code>Fn::GetAtt</code> mengembalikan Amazon Resource Name (ARN) domain. Atribut <code>DomainArn</code> <code>Fn::GetAtt</code> tidak lagi digunakan.	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
		AWS::RDS::DBCluster Gunakan properti <code>DBClusterIdentifier</code> untuk menentukan pengidentifikasi kluster DB. AWS::RDS::DBCluster Gunakan properti <code>DBClusterIdentifier</code> untuk menentukan pengidentifikasi kluster DB. AWS::Redshift::Cluster Gunakan properti <code>ClusterIdentifier</code> untuk menentukan pengidentifikasi unik pada kluster. AWS::Route53::HealthCheck Dalam jenis HealthCheckConfig properti, gunakan <code>Regions</code> properti untuk menentukan wilayah dari mana Anda ingin memeriksa kesehatan Route 53 untuk memeriksa titik akhir yang ditentukan. AWS::SSM::Document Gunakan properti <code>Tags</code> untuk menentukan tanda sumber daya CloudFormation untuk diterapkan ke dokumen.	
Sumber daya yang diperbarui	19 Februari 2018	AWS::CodeBuild::Project Gunakan properti <code>Triggers</code> untuk mengonfigurasi webhook agar proyek mulai membangun ulang kode sumber secara otomatis setiap kali perubahan kode didorong ke repositori. Ini hanya tersedia untuk GitHub proyek di CloudFormation. Ini tidak tersedia untuk proyek GitHub Enterprise.	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Sumber daya yang diperbarui	8 Februari 2018	AWS::DynamoDB::Table Gunakan properti <code>SSESpecification</code> untuk menentukan pengaturan untuk mengaktifkan enkripsi sisi server.	2010-05-15
Sumber daya yang diperbarui	5 Februari 2018	AWS::CodeBuild::Project Dalam tipe properti <code>CodeBuild Project Source</code> Sumber : - Gunakan properti <code>GitCloneDepth</code> untuk menentukan kedalaman sejarah untuk mengunduh. - Gunakan <code>InsecureSsl</code> properti untuk menentukan apakah akan mengabaikan peringatan SSL saat menghubungkan ke repositori proyek GitHub Enterprise Anda.	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Sumber daya yang diperbarui	23 Januari 2018	AWS::AutoScaling::LifecycleHook Gunakan properti <code>LifecycleHookName</code> untuk menentukan nama hook siklus aktif. AWS::DynamoDB::Table Properti <code>AttributeDefinitions</code> sekarang membutuhkan penggantian ketika diperbarui. AWS::EC2::Instance Gunakan properti <code>CreditSpecification</code> untuk menentukan opsi kredit untuk penggunaan CPU instans T2. Gunakan <code>ElasticGpuSpecifications</code> properti untuk menentukan Elastic GPUs, sumber daya GPU yang dapat Anda lampirkan ke instans Anda untuk mempercepat kinerja grafis aplikasi Anda. AWS::EC2::VPC Properti <code>InstanceTenancy</code> sekarang tidak memerlukan gangguan ketika diperbarui dari "dedicated" ke "default" . AWS::ECS::Service Gunakan <code>HealthCheckGracePeriodSeconds</code> properti untuk menentukan periode waktu, dalam hitungan detik, bahwa penjadwal layanan Amazon ECS mengabaikan pemeriksaan kesehatan target Elastic Load Balancing yang tidak sehat setelah tugas pertama kali dimulai.	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
		AWS::IoT::TopicRule Dalam tipe properti DynamoDBAction, properti <code>RangeKeyField</code> dan <code>RangeKeyValue</code> tidak lagi diperlukan. AWS::KinesisAnalytics::ApplicationOutput Dalam tipe ApplicationOutput properti, gunakan <code>LambdaOutput</code> properti untuk mengidentifikasi fungsi Lambda sebagai tujuan saat mengkonfigurasi output aplikasi. AWS::Kinesis::Stream Gunakan <code>StreamEncryption</code> properti untuk mengaktifkan atau memperbarui enkripsi sisi server menggunakan aliran AWS KMS key untuk tertentu. AWS::Lambda::Function Gunakan properti <code>ReservedConcurrentExecutions</code> untuk menentukan maksimal eksekusi bersamaan yang Anda inginkan untuk fungsi tersebut. AWS::RDS::DBSubnetGroup Gunakan properti <code>DBSubnetGroupName</code> untuk menentukan nama untuk Grup Subnet DB. AWS::S3::Bucket Gunakan <code>BucketEncryption</code> properti untuk menentukan enkripsi default untuk bucket menggunakan enkripsi sisi server dengan kunci terkelola Amazon S3 SSE-S3 atau bucket (SSE-KMS). AWS KMS keys	

Perubahan	Tanggal rilis	Deskripsi	Versi API
		Dalam tipe properti ReplicationRule, gunakan properti <code>SourceSelectionCriteria</code> untuk menentukan filter tambahan dalam mengidentifikasi objek sumber yang ingin Anda replikasi. Dalam tipe properti ReplicationDestination: - Gunakan <code>AccessControlTranslation</code> properti untuk menentukan kepemilikan replika AWS akun yang memiliki bucket tujuan. - Gunakan properti <code>Account</code> untuk menentukan ID akun pemilik bucket tujuan. - Gunakan properti <code>EncryptionConfiguration</code> untuk menentukan informasi terkait enkripsi untuk bucket yang menjadi tujuan objek yang direplikasi. AWS::SSM::Association Gunakan <code>AssociationName</code> properti untuk menentukan nama asosiasi antara dokumen SSM dan EC2 instance yang berisi agen konfigurasi untuk memproses dokumen.	
Pemicu rollback ditambahkan ke konsol. CloudFormation	15 Januari 2018	Pemicu rollback memungkinkan Anda untuk CloudFormation memantau status aplikasi Anda selama pembuatan dan pembaruan tumpukan, dan untuk memutar kembali operasi itu jika aplikasi melanggar ambang batas alarm apa pun yang telah Anda tentukan. Untuk informasi selengkapnya, lihat Memantau dan Mengembalikan Operasi Tumpukan.	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Sumber daya yang diperbarui	12 Januari 2018	AWS::SSM::Parameter Gunakan properti <code>AllowedPattern</code> untuk menentukan ekspresi reguler yang digunakan untuk memvalidasi nilai parameter.	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Sumber daya baru	5 Desember 2017	AWS::Inspector::AssessmentTarget Gunakan sumber daya <code>AWS::Inspector::AssessmentTarget</code> untuk membuat target penilaian Amazon Inspector. AWS::Inspector::AssessmentTemplate Gunakan sumber daya <code>AWS::Inspector::AssessmentTemplate</code> untuk membuat templat penilaian Amazon Inspector. AWS::Inspector::ResourceGroup Gunakan sumber daya <code>AWS::Inspector::ResourceGroup</code> untuk membuat grup sumber daya Amazon Inspector, yang menentukan tanda yang mengidentifikasi sumber daya AWS yang membentuk target penilaian Amazon Inspector. AWS::ServiceDiscovery::Instance Gunakan <code>AWS::ServiceDiscovery::Instance</code> sumber daya untuk menentukan informasi tentang instance yang dibuat Amazon Route 53. AWS::ServiceDiscovery::PrivateDnsNamespace Gunakan <code>AWS::ServiceDiscovery::PrivateDnsNamespace</code> sumber daya untuk menentukan informasi tentang namespace pribadi untuk Amazon Route 53. AWS::ServiceDiscovery::PublicDnsNamespace Gunakan <code>AWS::ServiceDiscovery::PublicDnsNamespace</code> sumber daya untuk menentukan informasi tentang namespace publik untuk Amazon Route 53.	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
		AWS::ServiceDiscovery::Service Gunakan <code>AWS::ServiceDiscovery::Service</code> sumber daya untuk menentukan templat hingga lima catatan dan pemeriksaan kesehatan opsional yang ingin dibuat Amazon Route 53 saat mendaftarkan instance.	
Sumber daya yang diperbarui	5 Desember 2017	AWS::KinesisAnalytics::Application Dalam tipe Input properti, gunakan <code>InputProcessingConfiguration</code> properti untuk mengubah catatan saat diterima dari aliran.	2010-05-15
Sumber daya yang diperbarui	1 Desember 2017	AWS::CodeBuild::Project Gunakan properti <code>BadgeEnabled</code> untuk menghasilkan URL yang dapat diakses secara publik untuk rencana bangunan proyek. Gunakan properti <code>Cache</code> untuk mengonfigurasi pengaturan cache untuk dependensi bangunan. Gunakan <code>VpcConfig</code> properti untuk mengaktifkan CodeBuild akses sumber daya di VPC Amazon. Dalam tipe properti EnvironmentVariable , gunakan properti <code>Type</code> untuk menentukan jenis variabel lingkungan.	2010-05-15
Sumber daya baru	30 November 2017	AWS::Cloud9::EnvironmentEC2 Gunakan <code>AWS::Cloud9::EnvironmentEC2</code> sumber daya untuk membuat lingkungan EC2 pengembangan Amazon di AWS Cloud9.	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Sumber daya yang diperbarui	29 November 2017	AWS::ECS::TaskDefinition Gunakan properti Cpu untuk menentukan jumlah unit cpu yang diperlukan untuk tugas. Gunakan properti ExecutionRoleArn untuk menentukan ARN peran eksekusi. Gunakan properti Memory untuk menentukan jumlah (di MiB) memori yang diperlukan untuk tugas. Gunakan properti RequiresCompatibilities untuk menentukan tipe peluncuran yang diperlukan tugas. AWS::ECS::Service Gunakan properti LaunchType untuk menentukan tipe peluncuran untuk menjalankan layanan Anda. Gunakan properti NetworkConfiguration untuk menentukan konfigurasi jaringan untuk layanan. Gunakan properti PlatformVersion untuk menentukan versi platform untuk menjalankan layanan Anda.	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Sumber daya baru	28 November 2017	AWS::GuardDuty::Detector Gunakan <code>AWS::GuardDuty::Detector</code> sumber daya untuk membuat satu GuardDuty detektor Amazon. AWS::GuardDuty::IPSet Gunakan <code>AWS::GuardDuty::IPSet</code> sumber daya untuk membuat set GuardDuty IP Amazon. AWS::GuardDuty::ThreatIntelSet Gunakan <code>AWS::GuardDuty::ThreatIntelSet</code> sumber daya untuk membuat file ThreatIntelSet.	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Sumber daya yang diperbarui	28 November 2017	AWS::CodeDeploy::Application Gunakan <code>ComputePlatform</code> properti untuk menentukan platform AWS Lambda komputasi CodeDeploy untuk menyebarkan aplikasi ke. AWS::CodeDeploy::DeploymentGroup Dalam tipe DeploymentStyle properti, gunakan <code>DeploymentType</code> properti untuk menentukan penerapan biru/hijau pada platform komputasi Lambda. AWS::EC2::SpotFleet Dalam tipe properti SpotFleetRequestConfigData, properti <code>SpotPrice</code> sekarang opsional. AWS::Lambda::Alias Gunakan <code>RoutingConfig</code> properti untuk menentukan dua versi AWS Lambda fungsi yang berbeda, memungkinkan Anda menentukan berapa persentase lalu lintas yang akan memanggil setiap versi.	2010-05-15
Kebijakan pembaruan <code>CodeDeployLambdaAliasUpdate</code> baru	28 November 2017	Gunakan kebijakan <code>CodeDeployLambdaAliasUpdate</code> pembaruan untuk melakukan CodeDeploy penerapan saat versi berubah pada <code>AWS::Lambda::Alias</code> sumber daya. Untuk informasi selengkapnya, lihat UpdatePolicyAtribut .	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Tipe parameter SSM baru	21 November 2017	Gunakan tipe SSM parameter untuk menggunakan parameter yang ada dari Systems Manager Parameter Store. Catatan: CloudFormation saat ini tidak mendukung tipe <code>SecureString</code> . Untuk informasi selengkapnya, lihat Tipe Parameter SSM .	2010-05-15
Bidang <code>ResolvedValue</code> baru untuk tipe data <code>Parameter</code>	21 November 2017	Bidang <code>ResolvedValue</code> mengembalikan nilai yang digunakan dalam definisi tumpukan untuk parameter SSM. Untuk informasi selengkapnya, lihat tipe Parameter data di Referensi AWS CloudFormation API.	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Sumber daya yang diperbarui	20 November 2017	AWS::ApiGateway::ApiKey Gunakan properti <code>CustomerId</code> untuk menentukan pengidentifikasi pelanggan Marketplace AWS . Gunakan properti <code>GenerateDistinctId</code> untuk menentukan apakah pengidentifikasi kunci berbeda dari nilai kunci API yang dibuat. AWS::ApiGateway::Authorizer Gunakan properti <code>AuthType</code> untuk menentukan bidang yang ditetapkan pelanggan yang digunakan dalam impor dan ekspor Swagger tanpa dampak fungsional. AWS::ApiGateway::DomainName Gunakan <code>EndpointConfiguration</code> properti untuk menentukan tipe titik akhir dari nama domain API Gateway. Gunakan properti <code>RegionalCertificateArn</code> untuk mereferensikan sertifikat untuk digunakan oleh titik akhir wilayah untuk nama domain. AWS::ApiGateway::Method Dalam tipe properti Integration dan IntegrationResponse, gunakan properti <code>ContentHandling</code> untuk menentukan cara menangani konversi tipe konten muatan permintaan. AWS::ApiGateway::RestApi Gunakan <code>EndpointConfiguration</code> properti untuk menentukan tipe titik akhir API Gateway REST API.	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
		AWS::ApplicationAutoScaling::ScalableTarget Gunakan <code>ScheduledActions</code> properti untuk menentukan tindakan terjadwal untuk target scalable Application Auto Scaling. AWS::ECR::Repository Gunakan <code>LifecyclePolicy</code> properti untuk menentukan kebijakan siklus hidup untuk repositori Amazon ECR. AWS::ECS::TaskDefinition Dalam jenis ContainerDefinition properti, gunakan <code>LinuxParameters</code> properti untuk menentukan opsi khusus Linux untuk wadah Amazon ECS. AWS::ElastiCache::ReplicationGroup Gunakan properti <code>AtRestEncryptionEnabled</code> untuk mengaktifkan enkripsi saat istirahat. Gunakan properti <code>AuthToken</code> untuk menentukan kata sandi yang digunakan untuk mengakses server yang dilindungi dengan kata sandi. Gunakan properti <code>TransitEncryptionEnabled</code> untuk mengaktifkan enkripsi dalam transit. AWS::ElasticLoadBalancingV2::TargetGroup Gunakan <code>TargetGroupName</code> atribut dengan <code>Fn::GetAtt</code> fungsi untuk mendapatkan nama kelompok target Elastic Load Balancing. AWS::Elasticsearch::Domain Gunakan <code>VPCOptions</code> properti untuk menentukan konfigurasi VPC untuk domain OpenSearch Layanan.	

Perubahan	Tanggal rilis	Deskripsi	Versi API
		AWS::EMR::Cluster Gunakan <code>EbsRootVolumeSize</code> properti untuk menentukan ukuran volume root EBS untuk klaster EMR Amazon. AWS::RDS::DBInstance Gunakan properti <code>SourceRegion</code> dan <code>KmsKeyId</code> untuk membuat replika baca terenkripsi dari instans DB lintas-wilayah. AWS::Route53::HostedZone Gunakan properti <code>QueryLoggingConfig</code> untuk menentukan konfigurasi untuk pencatatan kueri DNS.	
Bidang NoEcho baru untuk objek Response sumber daya kustom	20 November 2017	Anda sekarang dapat menggunakan bidang <code>NoEcho</code> opsional untuk menutupi output sumber daya kustom. Untuk informasi selengkapnya, lihat Objek Respons Sumber Daya Kustom. Parameter <code>noEcho</code> yang sesuai didukung oleh metode <code>send</code>. Untuk informasi selengkapnya, lihat Modul <code>cfn-response</code>.	2010-05-15
Penimpaan instans tumpukan ditambahkan untuk set tumpukan.	17 November 2017	CloudFormation StackSets memungkinkan Anda untuk mengganti nilai parameter dalam instance tumpukan berdasarkan akun dan wilayah. Anda dapat menimpa nilai parameter ketika Anda membuat instans tumpukan, atau ketika memperbarui instans tumpukan yang ada. Untuk informasi selengkapnya, lihat Parameter Penimpaan pada Instans Tumpukan .	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Sumber daya yang diperbarui	15 November 2017	AWS::StepFunctions::StateMachine Anda dapat menggunakan <code>AWS::StepFunctions::StateMachine</code> untuk menentukan <code>StateMachineName</code> saat membuat mesin status, dan keduanya <code>DefinitionString</code> dan <code>RoleArn</code> dapat diperbarui tanpa mengganti mesin status.	2010-05-15
StackSets sekarang mendukung maksimum 500 instance tumpukan per set tumpukan.	6 November 2017	Anda sekarang dapat membuat hingga maksimal 500 instans tumpukan per set tumpukan. Untuk informasi selengkapnya tentang AWS CloudFormation batas, lihat Memahami CloudFormation kuota .	2010-05-15
Sumber daya baru	2 November 2017	AWS::CloudFront::CloudFrontOriginAccessIdentity Gunakan <code>AWS::CloudFront::CloudFrontOriginAccessIdentity</code> sumber daya untuk menentukan identitas akses CloudFront asal Amazon untuk dikaitkan dengan asal CloudFront distribusi. AWS::CloudFront::StreamingDistribution Gunakan <code>AWS::CloudFront::StreamingDistribution</code> sumber daya untuk menentukan distribusi streaming Adobe Real-Time Messaging Protocol (RTMP). CloudFront	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Sumber daya yang diperbarui	2 November 2017	AWS::ApiGateway::Deployment Properti <code>StageName</code> tidak lagi digunakan di tipe properti StageDescription. AWS::ApiGateway::Method Gunakan <code>OperationName</code> properti untuk menetapkan nama ramah ke metode API Gateway. Gunakan properti <code>RequestValidatorId</code> untuk menghubungkan validator permintaan dengan metode. AWS::AutoScaling::AutoScalingGroup Gunakan <code>LifecycleHookSpecificationList</code> properti untuk menentukan tindakan yang akan dilakukan saat Auto Scaling meluncurkan atau menghentikan instance. AWS::CloudFront::Distribution Gunakan <code>Tags</code> properti untuk menentukan kumpulan tag arbitrer (pasangan kunci-nilai) untuk dikaitkan dengan distribusi. CloudFront Dalam tipe CacheBehavior dan DefaultCacheBehavior properti, gunakan <code>LambdaFunctionAssociations</code> properti untuk menentukan asosiasi fungsi Lambda untuk distribusi. CloudFront Dalam tipe properti CustomOriginConfig, gunakan properti <code>OriginKeepaliveTimeout</code> untuk menentukan batas waktu tetap aktif khusus, dan gunakan properti <code>OriginReadTimeout</code> untuk menentukan batas waktu baca asal kustom.	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
		Dalam jenis DistributionConfig properti, gunakan <code>IPV6Enabled</code> properti untuk menentukan apakah CloudFront merespons permintaan IPv6 DNS dengan IPv6 alamat untuk distribusi Anda. AWS::CodeDeploy::DeploymentGroup Dalam tipe LoadBalancerInfo properti, gunakan <code>TargetGroupInfoList</code> properti untuk menentukan informasi tentang grup target di Elastic Load Balancing yang akan digunakan dalam penerapan. AWS::EC2::SecurityGroup, AWS::EC2::SecurityGroupEgress, dan AWS::EC2::SecurityGroupIngress Gunakan properti <code>Description</code> untuk menentukan deskripsi aturan grup keamanan. AWS::EC2::Subnet Properti <code>Ipv6CidrBlock</code> sekarang mendukung pembaruan <code>No interruption</code> . AWS::EC2::VPNGateway Gunakan properti <code>AmazonSideAsn</code> untuk menentukan Autonomous System Number (ASN) privat untuk sisi Amazon sesi BGP. AWS::EC2::VPNConnection Gunakan properti <code>VpnTunnelOptionsSpecifications</code> untuk mengkonfigurasi opsi terowongan untuk koneksi VPN.	

Perubahan	Tanggal rilis	Deskripsi	Versi API
		AWS::ElasticBeanstalk::ConfigurationTemplate dan AWS::ElasticBeanstalk::Environment Dalam tipe properti ConfigurationOptionSetting dan OptionSetting, gunakan properti <code>ResourceName</code> untuk menentukan nama sumber daya untuk opsi konfigurasi penskalaan berbasis waktu. AWS::EMR::Cluster Gunakan properti <code>CustomAmiId</code> untuk menentukan AMI Linux Amazon kustom untuk klaster. AWS::KinesisFirehose::DeliveryStream Gunakan atribut <code>Arn</code> dengan fungsi <code>Fn::GetAtt</code> untuk mendapatkan Amazon Resource Name (ARN) pada aliran pengiriman. AWS::KMS::Key Gunakan <code>Tags</code> properti untuk menentukan kumpulan tag arbitrer (pasangan kunci-nilai) untuk dikaitkan dengan kunci yang dikelola pelanggan. AWS::OpsWorks::Layer dan AWS::OpsWorks::Stack Gunakan <code>Tags</code> properti untuk menentukan kumpulan tag arbitrer (pasangan kunci-nilai) untuk dikaitkan dengan AWS OpsWorks lapisan atau tumpukan. AWS::RDS::OptionGroup Dalam tipe properti OptionConfiguration, gunakan properti <code>OptionVersion</code> untuk menentukan versi untuk opsi.	

Perubahan	Tanggal rilis	Deskripsi	Versi API
		AWS::S3::Bucket Gunakan <code>AnalyticsConfigurations</code> properti untuk mengonfigurasi filter analisis untuk bucket Amazon S3.	

Perubahan	Tanggal rilis	Deskripsi	Versi API
Sumber daya baru	24 Oktober 2017	AWS::Glue::Classifier Gunakan sumber daya <code>AWS::Glue::Classifier</code> untuk membuat pengklasifikasi AWS Glue . AWS::Glue::Connection Gunakan <code>AWS::Glue::Connection</code> sumber daya untuk menentukan AWS Glue koneksi ke sumber data. AWS::Glue::Crawler Gunakan sumber daya <code>AWS::Glue::Crawler</code> untuk menentukan crawler AWS Glue . AWS::Glue::Database Gunakan sumber daya <code>AWS::Glue::Database</code> untuk membuat basis data AWS Glue . AWS::Glue::DevEndpoint Gunakan sumber daya <code>AWS::Glue::DevEndpoint</code> untuk menentukan titik akhir pengembangan untuk debugging skrip ETL dari jarak jauh. AWS::Glue::Job Gunakan sumber daya <code>AWS::Glue::Job</code> untuk menentukan tugas AWS Glue dalam katalog data. AWS::Glue::Partition Gunakan sumber daya <code>AWS::Glue::Partition</code> untuk membuat partisi AWS Glue , yang mewakili potongan data tabel. AWS::Glue::Table Gunakan sumber daya <code>AWS::Glue::Table</code> untuk membuat tabel AWS Glue .	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
		AWS::Glue::Trigger Gunakan <code>AWS::Glue::Trigger</code> sumber daya untuk menentukan pemicu yang menjalankan AWS Glue pekerjaan.	
Sumber daya baru	11 Oktober 2017	AWS::SSM::MaintenanceWindow Gunakan <code>AWS::SSM::MaintenanceWindow</code> sumber daya untuk membuat Jendela AWS Systems Manager Pemeliharaan. AWS::SSM::MaintenanceWindowTarget Gunakan sumber daya <code>AWS::SSM::MaintenanceWindowTarget</code> untuk mendaftarkan target dengan Jendela Pemeliharaan. AWS::SSM::MaintenanceWindowTask Gunakan sumber daya <code>AWS::SSM::MaintenanceWindowTask</code> untuk menetapkan tugas Jendela Pemeliharaan. AWS::SSM::PatchBaseline Gunakan <code>AWS::SSM::PatchBaseline</code> sumber daya untuk menentukan baseline patch Systems Manager.	2010-05-15
Sumber daya baru	10 Oktober 2017	AWS::ElasticLoadBalancingV2::ListenerCertificate Gunakan <code>AWS::ElasticLoadBalancingV2::ListenerCertificate</code> sumber daya untuk menentukan sertifikat untuk pendengar Elastic Load Balancing.	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Sumber daya baru	27 September 2017	AWS::Athena::NamedQuery Gunakan <code>AWS::Athena::NamedQuery</code> sumber daya untuk membuat kueri Amazon Athena.	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Sumber daya yang diperbarui	27 September 2017	AWS::EC2::NatGateway Gunakan properti Tags untuk menentukan tanda sumber daya untuk gateway NAT. AWS::ElasticBeanstalk::Application Gunakan ResourceLifecycleConfig properti untuk menentukan setelan siklus hidup untuk sumber daya milik aplikasi, dan peran layanan yang diasumsikan Elastic Beanstalk untuk menerapkan pengaturan siklus hidup. AWS::ElasticBeanstalk::ConfigurationTemplate dan AWS::ElasticBeanstalk::Environment Gunakan PlatformArn properti untuk menentukan platform khusus untuk Elastic Beanstalk. AWS::ElasticLoadBalancingV2::TargetGroup Dalam jenis TargetDescription properti, gunakan AvailabilityZone properti untuk menentukan Availability Zone tempat alamat IP akan didaftarkan. AWS::Events::Rule Dalam tipe properti Target, gunakan properti berikut untuk transformasi input peristiwa dan menyetel tugas Amazon ECS dan target aliran Kinesis. • EcsParameters • InputTransformer • KinesisParameters • RunCommandParameters AWS::KinesisFirehose::DeliveryStream Gunakan DeliveryStreamType properti untuk menentukan jenis aliran dan KinesisSt	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
		<code>reamSourceConfiguration</code> properti untuk menentukan aliran dan peran ARNs untuk aliran Kinesis yang digunakan sebagai sumber untuk aliran pengiriman. AWS::RDS::DBInstance Untuk properti Engine, jika Anda telah menentukan <code>oracle-se</code> atau <code>oracle-se1</code>, Anda dapat memperbarui ke <code>oracle-se2</code> tanpa mengganti instans basis data. AWS::S3::Bucket Gunakan <code>AccelerateConfiguration</code> properti untuk mengonfigurasi status akselerasi transfer untuk bucket Amazon S3.	
Perlindungan penghentian ditambahkan untuk tumpukan.	26 September 2017	Mengaktifkan perlindungan penghentian pada tumpukan mencegahnya dihapus secara tidak sengaja. Pengguna tidak dapat menghapus tumpukan dengan perlindungan terminasi diaktifkan. Untuk informasi selengkapnya, lihat Melindungi Tumpukan Agar Tidak Dihapus.	2010-05-15
Mengubah nilai umask default dari versi 1.4-22 dan seterusnya	14 September 2017	Nilai parameter umask default untuk file konfigurasi <code>cfn-hup.conf</code> sekarang adalah <code>022</code>. Untuk informasi selengkapnya, lihat cfn-hup.	

Perubahan	Tanggal rilis	Deskripsi	Versi API
Sumber daya yang diperbarui	7 September 2017	AWS::ElasticLoadBalancingV2::LoadBalancer Gunakan SubnetMappings properti untuk menentukan subnet untuk dilampirkan ke penyeimbang beban. IDs Gunakan properti Type untuk menentukan tipe penyeimbang beban yang akan dibuat. AWS::ElasticLoadBalancingV2::TargetGroup Gunakan properti TargetType untuk menentukan tipe registrasi target dalam grup target ini.	2010-05-15
Pemicu rollback ditambahkan ke API CloudFormation	31 Agustus 2017	Pemicu rollback memungkinkan Anda untuk CloudFormation memantau status aplikasi Anda selama pembuatan dan pembaruan tumpukan, dan untuk memutar kembali operasi itu jika aplikasi melanggar ambang batas alarm apa pun yang telah Anda tentukan. Untuk informasi selengkapnya, lihat RollbackConfiguration di dalam Referensi API AWS CloudFormation .	2010-05-15
Parameter umask baru untuk file cfn-hup.conf	31 Agustus 2017	Gunakan parameter umask dalam file konfigurasi cfn-hup.conf untuk mengendalikan izin file yang digunakan oleh daemon cfn-hup (versi 1.4-21). Untuk informasi selengkapnya, lihat cfn-hup .	

Perubahan	Tanggal rilis	Deskripsi	Versi API
Sumber daya yang diperbarui untuk dukungan Pengukuran VPC	29 Agustus 2017	AWS::EC2::VPCCidrBlock Gunakan <code>CidrBlock</code> properti untuk mengaitkan blok IPv4 CIDR dengan VPC. AWS::EC2::VPC Gunakan <code>CidrBlockAssociations</code> atribut dengan <code>Fn::GetAtt</code> fungsi untuk mendapatkan daftar asosiasi blok IPv4 CIDR yang IDs terkait dengan VPC.	2010-05-15
Sumber daya yang diperbarui	23 Agustus 2017	AWS::S3::Bucket Dalam tipe Rule properti, gunakan <code>TagFilters</code> properti untuk menentukan tag yang akan digunakan dalam mengidentifikasi subset objek untuk bucket Amazon S3. Gunakan <code>MetricsConfiguration</code> properti untuk menentukan konfigurasi metrik untuk metrik CloudWatch permintaan dari bucket Amazon S3. AWS::IoT::TopicRule Dalam tipe properti Action, gunakan properti <code>DynamoDBv2Action</code> untuk menjelaskan tindakan AWS IoT yang menulis data ke tabel DynamoDB. Dalam tipe properti Action, properti <code>DynamoDBAction</code> sekarang mendukung properti <code>HashKeyType</code> dan <code>RangeKeyType</code>. AWS::Lambda::Permission Gunakan properti <code>EventSourceToken</code> untuk menentukan token unik yang harus disediakan oleh prinsipal yang menjalankan fungsi tersebut.	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Parameter semu baru	23 Agustus 2017	Gunakan parameter semu <code>AWS::Partition</code> untuk mengembalikan partisi tempat sumber daya berada. Gunakan parameter semu <code>AWS::URLSuffix</code> untuk mengembalikan sufiks untuk domain. Untuk informasi selengkapnya, lihat Referensi Parameter Semu.	2010-05-15
Sumber daya baru untuk dukungan DAX	22 Agustus 2017	AWS::DAX::Cluster Gunakan <code>AWS::DAX::Cluster</code> sumber daya untuk membuat cluster DAX untuk digunakan dengan Amazon DynamoDB. AWS::DAX::ParameterGroup Gunakan <code>AWS::DAX::ParameterGroup</code> sumber daya untuk membuat grup parameter untuk digunakan dengan Amazon DynamoDB. AWS::DAX::SubnetGroup Gunakan sumber daya <code>AWS::DAX::SubnetGroup</code> untuk membuat grup subnet untuk digunakan dengan DAX (akselerator DynamoDB).	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Sumber daya baru	18 Agustus 2017	AWS::ApiGateway::DocumentationPart dan AWS::ApiGateway::DocumentationPart Gunakan <code>AWS::ApiGateway::DocumentationPart</code> sumber daya <code>AWS::ApiGateway::DocumentationPart</code> dan sumber daya untuk membuat dokumentasi API Gateway API Anda. AWS::ApiGateway::GatewayResponse Gunakan <code>AWS::ApiGateway::GatewayResponse</code> sumber daya untuk membuat respons khusus untuk API Gateway API Anda. AWS::ApiGateway::RequestValidator Gunakan <code>AWS::ApiGateway::RequestValidator</code> sumber daya untuk menyiapkan aturan validasi untuk permintaan masuk ke API Gateway API Anda. AWS::EC2::NetworkInterfacePermission Gunakan <code>AWS::EC2::NetworkInterfacePermission</code> sumber daya untuk memberikan izin AWS akun ke antarmuka jaringan.	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Sumber daya yang diperbarui	18 Agustus 2017	AWS::ApiGateway::Stage Gunakan properti <code>DocumentationVersion</code> untuk menentukan snapshot versioning dokumentasi API. AWS::AutoScaling::ScalingPolicy Gunakan <code>TargetTrackingConfiguration</code> properti untuk menentukan konfigurasi kebijakan penskalaan pelacakan target Auto Scaling. AWS::CloudTrail::Trail Gunakan <code>EventSelectors</code> properti untuk dukungan Amazon S3 Data Events. AWS::CodeDeploy::DeploymentGroup Gunakan <code>DeploymentStyle</code> properti <code>LoadBalancerInfo</code> dan untuk menentukan penyeimbang beban Elastic Load Balancing untuk penerapan di tempat. Gunakan properti <code>AutoRollbackConfiguration</code> untuk mengkonfigurasi rollback otomatis untuk deployment. AWS::EC2::SpotFleet Dalam tipe properti SpotFleetRequestConfigData, gunakan properti <code>ReplaceUnhealthyInstances</code> untuk menunjukkan apakah armada Spot harus menggantikan instans yang tidak sehat dan properti <code>Type</code> untuk menentukan tipe permintaan.	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
		AWS::EC2::Subnet Gunakan <code>Ipv6CidrBlock</code> properti <code>AssignIpv6AddressOnCreation</code> dan untuk membuat subnet dengan blok IPv6 CIDR. AWS::KinesisFirehose::DeliveryStream Gunakan <code>ExtendedS3DestinationConfiguration</code> properti untuk mengonfigurasi tujuan di Amazon S3. Gunakan subproperti <code>ProcessingConfiguration</code> dalam setiap konfigurasi tujuan untuk memanggil fungsi Lambda yang mentransformasikan data sumber yang masuk dan mengirimkan data yang ditransformasikan ke tujuan. AWS::RDS::DBCluster dan AWS::RDS::DBInstance <code>DeletionPolicy</code> Defaultnya sekarang <code>Snapshot</code> untuk <code>AWS::RDS::DBCluster</code> sumber <code>AWS::RDS::DBInstance</code> daya dan sumber daya yang tidak menentukan <code>DBClusterIdentifier</code> properti. Untuk informasi selengkapnya, lihat DeletionPolicyAtribut. AWS::S3::Bucket Dalam jenis Rule properti, gunakan <code>AbortIncompleteMultipartUpload</code> properti untuk menentukan aturan siklus hidup yang membatalkan unggahan multibagian yang tidak lengkap ke bucket Amazon S3.	

Perubahan	Tanggal rilis	Deskripsi	Versi API
		AWS::SQS::Queue Gunakan <code>KmsDataKeyReusePeriodSeconds</code> properti <code>KmsMasterKeyId</code> dan untuk mengonfigurasi enkripsi sisi server untuk Amazon SQS. Menambahkan atribut <code>Arn</code> untuk fungsi intrinsik <code>Fn::GetAtt</code> untuk sumber daya berikut: • AWS::CloudTrail::Trail. Juga ditambahkan <code>SnsTopicArn</code>. • AWS::CloudWatch::Alarm • AWS::DynamoDB::Table • AWS::ECS::Cluster • AWS::IoT::Policy • AWS::IoT::TopicRule • AWS::Logs::Destination	
Support untuk tag tumpukan di CodePipeline artefak	18 Agustus 2017	Anda sekarang dapat menentukan tag untuk tumpukan dalam file konfigurasi templat untuk digunakan sebagai artefak untuk CodePipeline saluran pipa. Tanda yang ditentukan diterapkan pada tumpukan yang dibuat menggunakan file konfigurasi templat. Untuk informasi lebih lanjut, lihat CloudFormation Artefak .	2010-05-15
Buat sistem file terenkripsi	14 Agustus 2017	AWS::EFS::FileSystem Gunakan <code>Encrypted</code> properti untuk mengenkripsi sistem file Amazon EFS selama pembuatan. Gunakan <code>KmsKeyId</code> properti untuk secara opsional menentukan kunci terkelola pelanggan khusus yang akan digunakan untuk melindungi sistem file terenkripsi.	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Sumber daya baru untuk AWS Batch dukungan	8 Agustus 2017	AWS::Batch::ComputeEnvironment Gunakan <code>AWS::Batch::ComputeEnvironment</code> sumber daya untuk menentukan lingkungan AWS Batch komputasi Anda. AWS::Batch::JobDefinition Gunakan <code>AWS::Batch::JobDefinition</code> sumber daya untuk menentukan parameter untuk definisi AWS Batch pekerjaan. AWS::Batch::JobQueue Gunakan <code>AWS::Batch::JobQueue</code> sumber daya untuk menentukan antrian AWS Batch pekerjaan Anda.	2010-05-15
Sumber daya baru untuk Amazon Managed Service untuk Apache Flink	28 Juli 2017	AWS::KinesisAnalytics::Application Gunakan <code>AWS::KinesisAnalytics::Application</code> sumber daya untuk membuat Amazon Managed Service untuk aplikasi Apache Flink. AWS::KinesisAnalytics::ApplicationOutput Gunakan <code>AWS::KinesisAnalytics::ApplicationOutput</code> sumber daya untuk menambahkan tujuan eksternal ke Amazon Managed Service untuk aplikasi Apache Flink Anda. AWS::KinesisAnalytics::ApplicationReferenceDataSource Gunakan <code>AWS::KinesisAnalytics::ApplicationReferenceDataSource</code> sumber daya untuk menambahkan sumber data referensi ke aplikasi Amazon Managed Service for Apache Flink yang ada.	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Gunakan StackSets untuk mengelola tumpukan secara terpusat di seluruh akun dan wilayah	25 Juli 2017	StackSets memungkinkan Anda membuat, memperbarui, atau menghapus tumpukan di beberapa akun dan wilayah dalam satu operasi. Dengan menggunakan akun administrator, Anda menentukan dan mengelola CloudFormation templat, dan menggunakan templat sebagai dasar untuk menyediakan tumpukan ke akun target yang dipilih di seluruh wilayah tertentu. Untuk informasi selengkapnya, lihat Mengelola tumpukan di seluruh akun dan Wilayah dengan StackSets .	2010-05-15
Lihat acara tumpukan berdasarkan token permintaan klien	14 Juli 2017	Dalam konsol, operasi tumpukan menampilkan token permintaan klien pada tab Acara. Semua acara yang dipicu oleh operasi tumpukan tertentu diberi token permintaan klien yang sama, yang dapat Anda gunakan untuk melacak operasi. Untuk informasi selengkapnya, lihat Melihat Data CloudFormation Tumpukan dan Sumber Daya di AWS Management Console dan StackEvent di Referensi AWS CloudFormation API.	2010-05-15
Gunakan tautan buat cepat tumpukan	14 Juli 2017	Gunakan tautan buat cepat untuk mendapatkan tumpukan dan berjalan dengan cepat. Anda dapat menentukan URL templat, nama tumpukan, dan parameter templat untuk mengisi terlebih dahulu satu halaman Buat Wizard Tumpukan. Untuk informasi selengkapnya, lihat Membuat Tautan Buat-Cepat untuk Tumpukan .	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Sumber daya baru untuk AWS Database Migration Service dukungan	12 Juli 2017	AWS::DMS::Certificate Gunakan <code>AWS::DMS::Certificate</code> sumber daya untuk membuat sertifikat SSL yang mengenkripsi koneksi antara AWS DMS titik akhir dan instance replikasi. AWS::DMS::Endpoint Gunakan sumber daya <code>AWS::DMS::Endpoint</code> untuk membuat titik akhir AWS DMS . AWS::DMS::EventSubscription Gunakan <code>AWS::DMS::EventSubscription</code> sumber daya untuk mendapatkan notifikasi AWS DMS acara melalui Amazon Simple Notification Service. AWS::DMS::ReplicationInstance Gunakan <code>AWS::DMS::ReplicationInstance</code> sumber daya untuk membuat contoh AWS DMS replikasi. AWS::DMS::ReplicationSubnetGroup Gunakan <code>AWS::DMS::ReplicationSubnetGroup</code> sumber daya untuk membuat grup subnet AWS DMS replikasi. AWS::DMS::ReplicationTask Gunakan <code>AWS::DMS::ReplicationTask</code> sumber daya untuk membuat tugas AWS DMS replikasi.	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Sumber daya baru	5 Juli 2017	AWS::CloudWatch::Dashboard Gunakan <code>AWS::CloudWatch::Dashboard</code> sumber daya untuk menentukan CloudWatch dasbor khusus untuk CloudWatch konsol Anda. AWS::ApiGateway::DomainName Gunakan <code>AWS::ApiGateway::DomainName</code> sumber daya untuk menentukan URL khusus dan ramah untuk API Anda yang diterapkan ke Amazon API Gateway. AWS::EC2::EgressOnlyInternetGateway Gunakan sumber daya <code>AWS::EC2::EgressOnlyInternetGateway</code> untuk membuat gateway internet khusus jalan keluar untuk VPC Anda. InstanceFleetConfig Gunakan <code>InstanceFleetConfig</code> sumber daya untuk mengonfigurasi armada Instans Spot untuk klaster EMR Amazon.	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Sumber daya yang diperbarui	5 Juli 2017	AWS::ApiGateway::RestApi Gunakan properti <code>BinaryMediaTypes</code> untuk menentukan tipe media biner yang didukung. AWS::ApplicationAutoScaling::ScalingPolicy Gunakan <code>TargetTrackingScalingPolicyConfiguration</code> properti untuk menentukan konfigurasi kebijakan penskalaan pelacakan target. AWS::CloudTrail::Trail Gunakan <code>TrailName</code> properti untuk menentukan nama kustom untuk AWS CloudTrail sumber daya. Gunakan properti <code>Tags</code> untuk menentukan tanda sumber daya. AWS::CodeDeploy::DeploymentGroup Gunakan properti <code>AlarmConfiguration</code> untuk mengkonfigurasi alarm untuk grup deployment. Gunakan properti <code>TriggerConfigurations</code> untuk mengkonfigurasi pemicu notifikasi untuk grup deployment. AWS::EMR::Cluster Gunakan <code>CoreInstanceFleet</code> properti dan properti dalam tipe <code>MasterInstanceFleet</code> JobFlowInstancesConfig properti untuk mengonfigurasi armada Instans Spot untuk klaster EMR Amazon.	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
		AWS::DynamoDB::Table Gunakan <code>TimeToLiveSpecification</code> properti untuk menentukan pengaturan Time to Live (TTL) untuk tabel Amazon DynamoDB. Gunakan <code>Tags</code> properti untuk menentukan tag sumber daya untuk tabel DynamoDB. AWS::EC2::Instance Properti <code>IamInstanceProfile</code> sekarang mendukung pembaruan <code>No interruption</code>. AWS::EC2::Route Gunakan <code>EgressOnlyInternetGatewayId</code> properti untuk menentukan gateway Internet khusus egres untuk rute. EC2 AWS::Kinesis::Stream Gunakan properti <code>RetentionPeriodHours</code> untuk menentukan jumlah jam catatan data yang disimpan dalam serpihan tetap dapat diakses. AWS::RDS::DBCluster Gunakan <code>ReplicationSourceIdentifier</code> properti untuk membuat cluster DB sebagai Replica Baca cluster DB lain atau instans Amazon RDS MySQL DB. AWS::Redshift::Cluster Gunakan <code>LoggingProperties</code> properti untuk membuat file log audit dan menyimpannya di Amazon S3.	

Perubahan	Tanggal rilis	Deskripsi	Versi API
Sumber daya baru	6 Juni 2017	AWS::EMR::SecurityConfiguration Gunakan sumber daya <code>AWS::EMR::SecurityConfiguration</code> untuk membuat konfigurasi keamanan, yang disimpan dalam layanan dan dapat ditentukan ketika klaster dibuat.	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Sumber daya yang diperbarui	6 Juni 2017	AWS::AutoScaling::LifecycleHook Properti NotificationTargetARN dan RoleARN sekarang opsional. AWS::CloudWatch::Alarm Sekarang Anda dapat menggunakan properti EvaluateLowSampleCountPercentile , ExtendedStatistic , dan TreatMissingData saat membuat sumber daya AWS::CloudWatch::Alarm . AWS::EC2::SpotFleet CloudFormation mendukung perubahan yang dapat diubah pada properti armada Spot. Properti properti SpotFleetRequestConfigurationData berikut mendukung pembaruan Replacement : • AllocationStrategy • IamFleetRole • LaunchSpecifications • SpotPrice • TerminateInstancesWithExpiration • ValidFrom • ValidUntil Properti properti SpotFleetRequestConfigurationData berikut mendukung pembaruan No interruption : • ExcessCapacityTerminationPolicy • TargetCapacity	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
		AWS::EMR::InstanceGroupConfig CloudFormation sekarang mendukung Auto Scaling untuk grup instans tugas Amazon EMR. AWS::Events::Rule Properti RoleArn tidak lagi digunakan pada sumber daya Rule. Gunakan RoleArn properti pada jenis Target properti untuk menentukan peran IAM yang akan digunakan untuk target. AWS::Kinesis::Stream Properti ShardCount sekarang mendukung pembaruan No interruption . AWS::Lambda::Function Gunakan TracingConfig properti untuk mengonfigurasi pengaturan penelusuran untuk fungsi Lambda. AWS::Redshift::Cluster, AWS::Redshift::ClusterParameterGroup, AWS::Redshift::ClusterSecurityGroup, dan AWS::Redshift::ClusterSubnetGroup Gunakan properti Tags untuk menentukan tanda sumber daya. AWS::RDS::DBCluster Menambahkan atribut ReadEndpoint.Address ke fungsi intrinsik Fn::GetAtt . AWS::S3::Bucket Menambahkan atribut Arn ke fungsi intrinsik Fn::GetAtt .	

Perubahan	Tanggal rilis	Deskripsi	Versi API
Sumber daya baru	11 Mei 2017	Sumber daya baru berikut mendukung penggunaan AWS WAF dengan Elastic Load Balancing (ELB) Application Load Balancers. AWS::WAFRegional::ByteMatchSet Gunakan sumber daya <code>AWS::WAFRegional::ByteMatchSet</code> untuk mengidentifikasi bagian permintaan web yang ingin Anda periksa. AWS::WAFRegional::IPSet Gunakan sumber daya <code>AWS::WAFRegional::IPSet</code> untuk menentukan permintaan web mana yang akan diizinkan atau diblokir berdasarkan alamat IP asal permintaan. AWS::WAFRegional::Rule Gunakan sumber daya <code>AWS::WAFRegional::Rule</code> untuk menentukan kombinasi objek <code>IPSet</code>, <code>ByteMatchSet</code>, dan <code>SqlInjectionMatchSet</code> yang mengidentifikasi permintaan web untuk mengizinkan, memblokir, atau menghitung. AWS::WAFRegional::SizeConstraintSet Gunakan sumber daya <code>AWS::WAFRegional::SizeConstraintSet</code> untuk menentukan batasan ukuran yang digunakan untuk memeriksa ukuran permintaan web dan bagian permintaan mana yang harus diperiksa. AWS::WAFRegional::SqlInjectionMatchSet Gunakan sumber daya <code>AWS::WAFRegional::SqlInjectionMatchSet</code> untuk mengizinkan, memblokir, atau menghitung permintaan yang	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
		berisi kode SQL berbahaya di bagian permintaan web tertentu. AWS::WAFRegional::WebACL Gunakan sumber daya <code>AWS::WAFRegional::WebACL</code> untuk mengidentifikasi permintaan web yang ingin Anda izinkan, blokir, atau hitung. AWS::WAFRegional::WebACLAssociation Gunakan sumber daya <code>AWS::WAFRegional::WebACLAssociation</code> untuk menghubungkan grup kontrol akses web (ACL) dengan sumber daya. AWS::WAFRegional::XssMatchSet Gunakan <code>AWS::WAFRegional::XssMatchSet</code> sumber daya untuk menentukan bagian permintaan web yang ingin Anda periksa AWS WAF untuk serangan skrip lintas situs dan nama header yang akan diperiksa.	

Perubahan	Tanggal rilis	Deskripsi	Versi API
Sumber daya baru	28 April 2017	AWS::Cognito::IdentityPool Gunakan <code>AWS::Cognito::IdentityPool</code> sumber daya untuk membuat kumpulan identitas Amazon Cognito. AWS::Cognito::IdentityPoolRoleAttachment Gunakan <code>AWS::Cognito::IdentityPoolRoleAttachment</code> sumber daya untuk mengelola konfigurasi peran untuk kumpulan identitas Amazon Cognito. AWS::Cognito::UserPool Gunakan <code>AWS::Cognito::UserPool</code> sumber daya untuk membuat kumpulan pengguna Amazon Cognito. AWS::Cognito::UserPoolClient Gunakan sumber daya <code>AWS::Cognito::UserPoolClient</code> untuk membuat klien kolam pengguna. AWS::Cognito::UserPoolGroup Gunakan <code>AWS::Cognito::UserPoolGroup</code> sumber daya untuk membuat grup pengguna di kumpulan pengguna Amazon Cognito. AWS::Cognito::UserPoolUser Gunakan <code>AWS::Cognito::UserPoolUser</code> sumber daya untuk membuat pengguna kumpulan pengguna Amazon Cognito.	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
		<u>AWS::Cognito::UserPoolUserToGroupAttachment</u> Gunakan <code>AWS::Cognito::UserPoolUserToGroupAttachment</code> sumber daya untuk melampirkan pengguna ke grup kumpulan pengguna Amazon Cognito.	

Perubahan	Tanggal rilis	Deskripsi	Versi API
Sumber daya yang diperbarui	28 April 2017	SourceDetails Gunakan subproperti <code>MaximumExecutionFrequency</code> sumber daya <code>AWS::Config::ConfigRule</code> untuk menjalankan evaluasi untuk aturan kustom menggunakan pemacu periodik. AWS::EC2::Volume Kami sekarang mendukung Volume Elastis untuk Amazon Elastic Block Store (Amazon EBS) di CloudFormation Kami sekarang mendukung pembaruan <code>No interruption</code> pada tiga properti: <code>VolumeType</code>, <code>Size</code>, dan <code>Iops</code>. AWS::EC2::SecurityGroup Gunakan <code>GroupName</code> properti untuk menentukan nama untuk grup EC2 keamanan Amazon Anda. AWS::ECS::Service Ada tiga properti baru untuk <code>AWS::ECS::Service</code>: <code>PlacementConstraints</code>, <code>PlacementStrategies</code>, dan <code>ServiceName</code>. AWS::ECS::TaskDefinition Gunakan properti <code>PlacementConstraints</code> untuk menentukan batasan penempatan untuk tugas dalam layanan. AWS::ElastiCache::ReplicationGroup Menambahkan atribut <code>ConfigurationEndpoint.Address</code> dan atribut <code>ConfigurationEndpoint.Port</code> untuk fungsi intrinsik <code>Fn::GetAtt</code>.	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
		<u>AWS::ElasticLoadBalancingV2::LoadBalancer</u> Gunakan properti <code>IpAddressType</code> untuk menentukan tipe alamat IP yang digunakan oleh subnet penyeimbang beban. <u>AWS::EMR::Cluster</u> CloudFormation sekarang mendukung Auto Scaling untuk cluster EMR Amazon. <u>AWS::IAM::ManagedPolicy</u> Gunakan <code>ManagedPolicyName</code> properti untuk menentukan nama kustom untuk kebijakan terkelola IAM Anda. <u>AWS::Lambda::Function</u> Gunakan <code>Tags</code> properti untuk menambahkan tag ke fungsi Lambda Anda. <u>AWS::OpsWorks::Instance</u> Menambahkan atribut berikut ke fungsi intrinsik <code>Fn::GetAtt : AvailabilityZone , PrivateDnsName , PrivateIp , dan PublicDnsName</code> . <u>AWS::OpsWorks::UserProfile</u> Gunakan properti <code>SshUsername</code> untuk menentukan nama SSH pengguna. Menambahkan atribut <code>SshUsername</code> ke fungsi intrinsik <code>Fn::GetAtt</code> . <u>AWS::Redshift::Cluster</u> Gunakan <code>IamRoles</code> properti untuk menyediakan daftar satu atau beberapa AWS Identity and Access Management peran yang dapat digunakan kluster	

Perubahan	Tanggal rilis	Deskripsi	Versi API
		Amazon Redshift untuk mengakses layanan lain AWS .	
Edit template di YAMAL dan JSON menggunakan Designer AWS CloudFormation	6 April 2017	Saat Anda membuat CloudFormation template menggunakan Designer, Anda sekarang dapat mengedit template Anda di YAMAL dan JSON di editor terintegrasi. Anda juga dapat mengkonversi templat JSON untuk YAMAL dan sebaliknya, bergantung pada bahasa penulisan templat pilihan Anda. Untuk informasi lebih lanjut, lihat Apa itu CloudFormation Designer? .	2010-05-15
Sumber daya baru	6 April 2017	AWS::SSM::Parameter Gunakan <code>AWS::SSM::Parameter</code> sumber daya untuk membuat parameter SSM di Parameter Store.	2010-05-15
Transformasi <code>AWS::Include</code>	28 Maret 2017	Gunakan <code>AWS::Include</code> transformasi untuk mereferensikan cuplikan yang dapat digunakan kembali yang disimpan dalam bucket Amazon S3. Untuk informasi selengkapnya, lihat AWS::IncludeMengubah .	2010-05-15
Rekan VPC Amazon Anda dengan akun lain	28 Maret 2017	Anda sekarang dapat menggunakan CloudFormation untuk mengintip VPC Amazon Anda dengan VPC di akun lain. AWS Untuk informasi selengkapnya, lihat Peer dengan VPC Amazon di AWS Akun Lain.	2010-05-15
Sumber daya baru	28 Maret 2017	AWS::ApiGateway::UsagePlanKey Gunakan sumber daya <code>AWS::ApiGateway::UsagePlanKey</code> untuk menghubungkan kunci rencana penggunaan dan menentukan pengguna mana yang menerapkan rencana penggunaan.	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Sumber daya yang diperbarui	28 Maret 2017	AWS::EC2::VPCPeeringConnection Gunakan properti <code>PeerOwnerId</code> dan properti <code>PeerRoleArn</code> untuk peer dengan VPC di akun AWS lain. Untuk informasi selengkapnya, lihat Peer dengan VPC Amazon di AWS Akun Lain. AWS::IAM::InstanceProfile Gunakan properti <code>InstanceProfileName</code> untuk mengkonfigurasi profil instans. AWS::Lambda::Function Gunakan properti <code>DeadLetterConfig</code> untuk mengkonfigurasi bagaimana AWS Lambda menangani acara yang tidak dapat diproses. Node.js v0.10 tidak lagi didukung untuk properti <code>Runtime</code>. AWS::Route53::HealthCheck Ada tujuh jenis subproperti sumber daya baru untuk HealthCheckConfig <code>HealthCheckConfig</code> properti: <code>AlarmIdentifier</code> , <code>ChildHealthChecks</code> , <code>EnableSNI</code> , <code>HealthThreshold</code> , <code>InsufficientDataHealthStatus</code> <code>Inverted</code> , dan <code>MeasureLatency</code> . AWS::SQS::Queue Gunakan <code>FifoQueue</code> properti <code>ContentBasedDeduplication</code> dan untuk membuat antrian Amazon Simple Queue Service First-In-First-Out (FIFO).	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
		AWS::S3::Bucket Anda sekarang dapat menentukan nama IPv6 domain untuk bucket Amazon S3 Anda.	
Sumber daya baru	10 Februari 2017	AWS::StepFunctions::Activity Gunakan <code>AWS::StepFunctions::Activity</code> sumber daya untuk membuat AWS Step Functions aktivitas. AWS::StepFunctions::StateMachine Gunakan <code>AWS::StepFunctions::StateMachine</code> sumber daya untuk membuat mesin status Step Functions.	2010-05-15
Fungsi intrinsik baru	17 Januari 2017	Gunakan fungsi <code>Fn::Split</code> untuk membagi string menjadi daftar nilai string. Untuk informasi selengkapnya, lihat Fn::Split .	2010-05-15
Dukungan konsol untuk membuat daftar impor	17 Januari 2017	Gunakan CloudFormation konsol untuk melihat semua tumpukan yang mengimpor nilai keluaran yang diekspor. Untuk informasi selengkapnya, lihat Membuat Daftar Tumpukan Yang Mengimpor Nilai Output yang Diekspor .	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Sumber daya yang diperbarui	17 Januari 2017	AWS::AutoScaling::AutoScalingGroup LoadBalancerNames Properti dapat diperbarui tanpa mengganti grup Auto Scaling. AWS::ECS::TaskDefinition Menambahkan properti NetworkMode dan MemoryReservation . AWS::RDS::DBCluster CloudFormation mendukung pembaruan ke Tags properti. AWS::RDS::DBInstance Properti Timezone yang ditambahkan. FirehoseAction Properti Separator yang ditambahkan. AWS::OpsWorks::Instance Menambahkan atribut PublicIp untuk atribut untuk fungsi intrinsik Fn::GetAtt .	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Sumber daya baru	01 Desember 2016	AWS::CodeBuild::Project Gunakan <code>AWS::CodeBuild::Project</code> sumber daya untuk membuat AWS CodeBuild proyek yang menentukan cara CodeBuild membangun kode sumber Anda. AWS::SSM::Association Gunakan <code>AWS::SSM::Association</code> sumber daya untuk mengaitkan dokumen Amazon EC2 Systems Manager dengan EC2 instans. AWS::EC2::SubnetCidrBlock Gunakan <code>AWS::EC2::SubnetCidrBlock</code> sumber daya untuk mengaitkan satu blok IPv6 CIDR dengan subnet Amazon VPC. AWS::EC2::VPCCidrBlock Gunakan <code>AWS::EC2::VPCCidrBlock</code> sumber daya untuk mengaitkan satu blok IPv6 CIDR yang disediakan Amazon dengan VPC Amazon.	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Sumber daya yang diperbarui untuk IPv6 dukungan	01 Desember 2016	AWS::EC2::Instance Menambahkan properti <code>Ipv6AddressCount</code> dan <code>Ipv6Addresses</code> . AWS::EC2::NetworkAclEntry Properti <code>Ipv6CidrBlock</code> yang ditambahkan. AWS::EC2::NetworkInterface Menambahkan properti <code>Ipv6AddressCount</code> dan <code>Ipv6Addresses</code> . AWS::EC2::Route Properti <code>DestinationIpv6CidrBlock</code> yang ditambahkan. AWS::EC2::SecurityGroupEgress Properti <code>CidrIpv6</code> yang ditambahkan. AWS::EC2::SecurityGroupIngress Properti <code>CidrIpv6</code> yang ditambahkan. AWS::EC2::SpotFleet Menambahkan properti <code>Ipv6AddressCount</code> dan <code>Ipv6Addresses</code> untuk antarmuka jaringan spesifikasi peluncuran. AWS::EC2::Subnet Menambahkan atribut <code>Ipv6CidrBlocks</code> untuk fungsi <code>Fn::GetAtt</code> . AWS::EC2::VPC Menambahkan atribut <code>Ipv6CidrBlocks</code> untuk fungsi <code>Fn::GetAtt</code> .	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
		AWS::SSM::Document Menambahkan properti DocumentType .	
Spesifikasi sumber daya	22 November 2016	Gunakan spesifikasi CloudFormation sumber daya untuk membuat alat yang membantu Anda membuat CloudFormation templat. Spesifikasinya adalah file teks berformat JSON yang dapat dibaca mesin. Untuk informasi selengkapnya, lihat Spesifikasi CloudFormation Sumber Daya .	2010-05-15
Sumber daya baru	22 November 2016	AWS::OpsWorks::UserProfile Gunakan <code>AWS::OpsWorks::UserProfile</code> sumber daya untuk mengonfigurasi akses SSH bagi pengguna yang memerlukan akses ke instance AWS OpsWorks dalam tumpukan. AWS::OpsWorks::Volume Gunakan <code>AWS::OpsWorks::Volume</code> sumber daya untuk mendaftarkan volume Amazon Elastic Block Store dengan AWS OpsWorks tumpukan.	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Sumber daya yang diperbarui	22 November 2016	AWS::OpsWorks::App Properti <code>DataSources</code> yang ditambahkan. AWS::OpsWorks::Instance Menambahkan properti <code>BlockDeviceMappings</code>, <code>AgentVersion</code>, <code>ElasticIps</code>, <code>Hostname</code>, <code>Tenancy</code>, dan <code>Volumes</code>. AWS::OpsWorks::Layer Menambahkan properti <code>CustomJson</code> dan <code>VolumeConfigurations</code>. AWS::OpsWorks::Stack Menambahkan properti <code>ElasticIps</code>, <code>EcsClusterArn</code>, <code>RdsDbInstances</code>, <code>CloneAppIds</code>, <code>ClonePermissions</code>, dan <code>SourceStackId</code>. AWS::RDS::DBInstance Menambahkan properti <code>CopyTagsToSnapshot</code>.	2010-05-15
Buat daftar impor	22 November 2016	Buat daftar impor dari nilai output yang diekspor untuk melacak CloudFormation tumpukan mana yang mengimpor nilai. Untuk informasi selengkapnya, lihat Membuat Daftar Tumpukan Yang Mengimpor Nilai Output yang Diekspor.	2010-05-15
Transformasi	17 November 2016	Tentukan Model Aplikasi AWS Tanpa Server (AWS SAM) yang CloudFormation digunakan untuk memproses sintaks AWS SAM untuk aplikasi tanpa server. Untuk informasi selengkapnya, lihat Transformasi.	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Sumber daya baru	17 November 2016	AWS::SNS::Subscription Gunakan <code>AWS::SNS::Subscription</code> sumber daya untuk berlangganan titik akhir ke topik Layanan Pemberitahuan Sederhana Amazon.	2010-05-15
Sumber daya yang diperbarui	17 November 2016	AWS::Lambda::Function Gunakan <code>Environment</code> properti untuk menentukan pasangan kunci-nilai (variabel lingkungan) yang dapat diakses oleh AWS Lambda fungsi Anda. Gunakan <code>KmsKeyArn</code> properti untuk menentukan kunci KMS yang AWS Lambda digunakan untuk mengenkripsi dan mendekripsi variabel lingkungan.	2010-05-15
Perintah CLI Baru	17 November 2016	Mengunggah Artefak Lokal ke Bucket S3 Gunakan package perintah untuk mengunggah artefak lokal yang direferensikan dalam CloudFormation template ke bucket S3. Menerapkan Template dengan Cepat dengan Transformasi Gunakan perintah <code>deploy</code> untuk menggabungkan tindakan membuat dan menjalankan set perubahan menjadi satu perintah. Perintah ini berguna untuk membuat atau memperbarui tumpukan dengan cepat yang berisi transformasi.	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Sumber daya yang diperbarui	03 November 2016	AWS::CloudFront::Distribution Untuk DistributionConfig properti, gunakan <code>HttpVersion</code> properti untuk menentukan versi HTTP terbaru yang dapat digunakan pemirsa untuk berkomunikasi dengan Amazon CloudFront. Untuk ForwardedValues properti, gunakan <code>QueryStringCacheKeys</code> properti untuk menentukan parameter string kueri yang CloudFront digunakan untuk menentukan konten mana yang akan di-cache.	2010-05-15
Buat daftar ekspor tumpukan	03 November 2016	Gunakan CloudFormation konsol, API, atau AWS CLI untuk melihat daftar semua nilai keluaran yang diekspor untuk suatu wilayah. Untuk informasi selengkapnya, lihat Mengekspor Nilai Output Tumpukan .	2010-05-15
Pengiriman terus menerus dengan tumpukan	03 November 2016	Gunakan AWS CodePipeline untuk membangun alur kerja pengiriman berkelanjutan dengan CloudFormation tumpukan. Untuk informasi selengkapnya, lihat Pengiriman Berkelanjutan dengan CodePipeline .	2010-05-15
Lewati sumber daya selama rollback	03 November 2016	Jika Anda memiliki tumpukan dalam status <code>UPDATE_ROLLBACK_FAILED</code> , gunakan parameter <code>ResourcesToSkip</code> untuk tindakan <code>ContinueUpdateRollback</code> untuk melewati sumber daya yang tidak dapat rollback oleh CloudFormation. Untuk informasi selengkapnya, lihat bagian Pemecahan Masalah di Rollback Pembaruan Gagal .	2010-05-15
Peningkatan set perubahan	03 November 2016	Anda dapat membuat tumpukan baru menggunakan set perubahan .	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Sumber daya yang diperbarui	12 Oktober 2016	AWS::ElastiCache::CacheCluster Perbarui properti <code>CacheNodeType</code> tanpa mengganti klaster. AWS::ElastiCache::ReplicationGroup Anda dapat membuat grup replikasi Redis (modus klaster diaktifkan) yang dapat berisi beberapa grup simpul (serpihan), masing-masing dengan klaster utama dan replika baca. AWS::ElastiCache::SubnetGroup Gunakan <code>CacheSubnetGroupName</code> properti untuk menentukan nama untuk grup ElastiCache subnet Amazon.	2010-05-15
Sumber daya baru	06 Oktober 2016	AWS::ApiGateway::UsagePlan Gunakan <code>AWS::ApiGateway::UsagePlan</code> sumber daya untuk menentukan paket penggunaan untuk Amazon API Gateway APIs yang diterapkan. AWS::CodeCommit::Repository Gunakan <code>AWS::CodeCommit::Repository</code> sumber daya untuk membuat CodeCommit repositori yang di-host oleh Amazon Web Services.	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Sumber daya yang diperbarui	06 Oktober 2016	AWS::ApiGateway::Authorizer Gunakan <code>ProviderARNs</code> properti untuk menggunakan kumpulan pengguna Amazon Cognito sebagai otorisasi API Amazon API Gateway. AWS::ApiGateway::Deployment Properti <code>StageName</code> tidak lagi diperlukan. AWS::ElasticLoadBalancingV2::TargetGroup Untuk <code>GetAtt</code> fungsi tersebut, gunakan <code>LoadBalancerArns</code> atribut untuk mengambil Amazon Resource Names (ARNs) dari penyeimbang beban yang merutekan lalu lintas ke grup target. AWS::RDS::DBInstance Gunakan properti <code>Domain</code> dan <code>DomainIAMRoleName</code> untuk menggunakan Otentikasi Windows saat pengguna terhubung ke instans DB RDS. AWS::EC2::SecurityGroupEgress Gunakan <code>DestinationPrefixListId</code> properti untuk menentukan awalan AWS layanan dari titik akhir VPC Amazon.	2010-05-15
Peningkatan referensi lintas-tupukan	06 Oktober 2016	Gunakan fungsi intrinsik untuk menyesuaikan nilai Name ekspor atau untuk merujuk ke nilai dalam fungsi ImportValue .	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
CloudFormation peran layanan	26 September 2016	Gunakan peran layanan AWS Identity and Access Management (IAM) untuk operasi CloudFormation tumpukan. CloudFormation menggunakan kredensial peran untuk melakukan panggilan untuk menumpuk sumber daya atas nama Anda. Untuk informasi selengkapnya, lihat peran AWS CloudFormation layanan .	2010-05-15
Fitur baru	19 September 2016	Anda dapat menggunakan bidang output Export dan fungsi intrinsik <code>Fn::ImportValue</code> agar satu tumpukan merujuk ke output sumber daya di tumpukan lain. Untuk informasi selengkapnya, lihat Output , Fn::ImportValue , dan Walkthrough: Lihat Output Sumber Daya di Tumpukan Lain . CloudFormation	2010-05-15
Dukungan YAML	19 September 2016	Anda dapat menggunakan format YAMAL untuk membuat CloudFormation template. YAML juga memungkinkan Anda untuk, misalnya, menambahkan komentar ke templat Anda atau menggunakan bentuk pendek untuk fungsi intrinsik. Untuk informasi selengkapnya, lihat format CloudFormation templat .	2010-05-15
Fungsi intrinsik baru	19 September 2016	Gunakan fungsi <code>Fn::Sub</code> untuk mengganti variabel dalam string input dengan nilai yang Anda tentukan. Untuk informasi selengkapnya, lihat Fn::Sub .	2010-05-15
Sumber daya baru	19 September 2016	AWS::KMS::Alias Gunakan <code>AWS::KMS::Alias</code> sumber daya untuk membuat alias untuk file AWS KMS key.	

Perubahan	Tanggal rilis	Deskripsi	Versi API
Sumber daya yang diperbarui	19 September 2016	AWS::EC2::SpotFleet Untuk properti <code>LaunchSpecifications</code> , gunakan properti <code>SpotPrice</code> untuk menentukan harga penawaran untuk tipe instans tertentu. AWS::ECS::Cluster Gunakan <code>ClusterName</code> properti untuk menentukan nama untuk klaster Amazon Elastic Container Service. AWS::ECS::TaskDefinition Gunakan <code>TaskRoleArn</code> properti untuk menentukan AWS Identity and Access Management peran yang digunakan penampung Amazon Elastic Container Service untuk melakukan AWS panggilan atas nama Anda. Gunakan properti <code>Family</code> untuk mendaftar ketentuan tugas untuk keluarga tertentu. AWS::Elasticsearch::Domain Gunakan <code>ElasticsearchVersion</code> properti untuk menentukan versi Elasticsearch mana yang akan digunakan.	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Sumber daya baru	11 Agustus, 2016	Gunakan sumber daya Elastic Load Balancing Application Load Balancer berikut untuk mendistribusikan lalu lintas aplikasi yang masuk ke beberapa target, EC2 seperti instance, di beberapa Availability Zone: • AWS::ElasticLoadBalancingV2::Listener • AWS::ElasticLoadBalancingV2::ListenerRule • AWS::ElasticLoadBalancingV2::LoadBalancer • AWS::ElasticLoadBalancingV2::TargetGroup	2010-05-15
Sumber daya yang diperbarui	11 Agustus, 2016	AWS::AutoScaling::AutoScalingGroup Gunakan TargetGroupARNs properti untuk mengaitkan grup Auto Scaling dengan satu atau beberapa grup target Application Load Balancer. AWS::ECS::Service Untuk LoadBalancers properti load, gunakan TargetGroupArn properti untuk mengaitkan layanan Amazon Elastic Container Service dengan grup target Application Load Balancer.	2010-05-15
Sumber daya baru	09 Agustus 2016	CloudFormation menambahkan sumber daya berikut: AWS::ApplicationAutoScaling::ScalableTarget dan AWS::ApplicationAutoScaling::ScalingPolicy Gunakan kebijakan penskalaan Application Auto Scaling untuk menentukan kapan dan bagaimana skala sumber daya target. AWS::CertificateManager::Certificate Berikan AWS Certificate Manager sertifikat yang dapat Anda gunakan dengan AWS layanan lain untuk mengaktifkan koneksi aman.	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Sumber daya yang diperbarui	09 Agustus 2016	CloudFormation memperbarui sumber daya berikut: AWS::CloudFront::Distribution Untuk properti <code>ViewerCertificate</code> konfigurasi distribusi, Anda dapat menentukan sertifikat AWS Certificate Manager . Untuk properti <code>Origin</code> konfigurasi distribusi, Anda dapat menentukan header kustom dan protokol SSL untuk asal kustom. AWS::EFS::FileSystem Anda dapat menentukan mode kinerja untuk sistem file Amazon Elastic File System.	2010-05-15
Sumber daya baru	20 Juli 2016	AWS IoT Gunakan AWS IoT untuk mendeklarasikan AWS IoT kebijakan, sertifikat X.509, hubungan antara kebijakan dan prinsipal (sertifikat X.509 atau kredensi lainnya), sesuatu, hubungan antara prinsipal dan AWS IoT sesuatu, atau aturan. AWS IoT • AWS::IoT::Certificate • AWS::IoT::Policy • AWS::IoT::PolicyPrincipalAttachment • AWS::IoT::Thing • AWS::IoT::ThingPrincipalAttachment • AWS::IoT::TopicRule	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Sumber daya yang diperbarui	20 Juli 2016	CloudFormation memperbarui sumber daya berikut: AWS::IAM::Group, AWS::IAM::Role, AWS::IAM::User Gunakan properti nama untuk menentukan nama kustom untuk sumber daya AWS Identity and Access Management (IAM). AWS::ApiGateway::Method Untuk <code>Integration</code> properti, Anda dapat menggunakan <code>PassthroughBehavior</code> properti untuk menentukan kapan Amazon API Gateway meneruskan permintaan ke back end yang ditargetkan. AWS::ApiGateway::Model dan AWS::ApiGateway::RestApi Anda dapat menentukan objek JSON untuk properti <code>Schema</code> dan <code>Body</code>.	2010-05-15
Grup Auto Scaling UpdatePolicy	9 Juni 2016	Untuk <code>UpdatePolicy</code> atribut, gunakan <code>AutoScalingReplacingUpdate</code> properti untuk menentukan apakah grup Auto Scaling dan instance yang dikandungnya diganti saat Anda memperbarui grup Auto Scaling. Selama penggantian, CloudFormation pertahankan grup Auto Scaling lama hingga berhasil membuat yang baru CloudFormation sehingga dapat memutar kembali ke grup Auto Scaling lama jika pembaruan gagal. Untuk informasi selengkapnya, lihat UpdatePolicy Atribut.	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Sumber daya baru	9 Juni 2016	CloudFormation menambahkan sumber daya berikut: AWS::EC2::FlowLog Membuat log aliran Amazon Elastic Compute Cloud yang menangkap lalu lintas IP untuk antarmuka jaringan, subnet, atau VPC tertentu. AWS::KinesisFirehose::DeliveryStream Membuat aliran pengiriman yang mengirimkan data streaming real-time ke suatu tujuan, seperti Amazon Simple Storage Service, Amazon Redshift, atau Amazon OpenSearch Service.	2010-05-15
Sumber daya yang diperbarui	9 Juni 2016	CloudFormation memperbarui sumber daya berikut: AWS::Kinesis::Stream Gunakan Name properti untuk menentukan nama untuk aliran Amazon Kinesis. AWS::Lambda::Function Untuk properti Code, Anda dapat menggunakan properti ZipFile dan modul respons cfn untuk lingkungan waktu aktif nodejs4.3 . AWS::SNS::Topic CloudFormation mengaktifkan pembaruan untuk sumber daya topik Layanan Pemberitahuan Sederhana Amazon.	2010-05-15
Sumber daya baru	25 April 2016	Gunakan AWS::EC2::Host sumber daya untuk mengalokasikan server fisik khusus untuk meluncurkan EC2 instance.	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Sumber daya yang diperbarui	25 April 2016	AWS::EC2::Instance Gunakan HostId properti Affinity dan untuk meluncurkan instans ke host khusus Amazon Elastic Compute Cloud. AWS::ECS::Service Gunakan properti DeploymentConfiguration untuk mengkonfigurasi berapa banyak tugas dapat berjalan selama deployment. AWS::ECS::TaskDefinition CloudFormation menambahkan dukungan untuk properti definisi container Amazon Elastic Container Service tambahan. AWS::GameLift::Fleet Gunakan MinSize properti MaxSize dan untuk menentukan jumlah EC2 instans maksimum dan minimum yang diizinkan dalam armada GameLift Server Amazon Anda. AWS::Lambda::Function Gunakan FunctionName properti untuk menentukan nama untuk AWS Lambda fungsi Anda. Anda juga dapat menggunakan Python 2.7 untuk menentukan fungsi sebaris.	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Sumber daya baru	18 April 2016	Amazon API Gateway Gunakan sumber daya Amazon API Gateway untuk mempublikasikan, memelihara, dan memantau APIs pada skala apa pun. Anda dapat membuat klien APIs dapat menelepon untuk mengakses layanan back-end Anda, seperti aplikasi yang menjalankan EC2 instance atau kode yang sedang berjalan. AWS Lambda • 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 Buat aturan CloudWatch Acara Amazon yang memantau perubahan AWS sumber daya di akun Anda (peristiwa). Jika peristiwa masuk cocok dengan kondisi yang Anda jelaskan dalam aturan, Amazon CloudWatch Events akan mengirim pesan ke dan mengaktifkan target yang Anda tentukan, seperti AWS Lambda fungsi atau topik Layanan Pemberitahuan Sederhana Amazon.	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
		AWS::WAF::SizeConstraintSet dan AWS::WAF::XssMatchSet Gunakan dua AWS WAF aturan untuk memeriksa ukuran permintaan web atau untuk mencegah serangan skrip lintas situs.	
Sumber daya baru	31 Maret 2016	Gunakan AWS::Lambda::Alias sumber daya untuk membuat alias untuk AWS Lambda fungsi Anda dan AWS::Lambda::Version sumber daya untuk membuat versi fungsi Anda.	2010-05-15
Sumber daya yang diperbarui	31 Maret 2016	CloudFormation memperbarui sumber daya berikut: AWS::EMR::Cluster dan AWS::EMR::InstanceGroupConfig Gunakan <code>EbsConfiguration</code> properti untuk mengonfigurasi volume penyimpanan Amazon Elastic Block Store untuk klaster EMR Amazon atau grup instans Anda. AWS::Lambda::Function Gunakan properti <code>VpcConfig</code> untuk mengaktifkan fungsi AWS Lambda untuk mengakses sumber daya dalam VPC. AWS::S3::Bucket Untuk aturan siklus hidup Amazon Simple Storage Service, Anda dapat menentukan beberapa aturan transisi yang menentukan kapan objek bertransisi ke kelas penyimpanan tertentu.	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Set perubahan	29 Maret 2016	Sebelum memperbarui tumpukan, gunakan set perubahan untuk melihat bagaimana perubahan Anda dapat memengaruhi sumber daya yang sedang berjalan. Untuk informasi selengkapnya, lihat Memperbarui Tumpukan Menggunakan Set Perubahan .	2010-05-15
Sumber daya baru	15 Maret 2016	Gunakan AWS::GameLift::Alias , AWS::GameLift::Build , dan AWS::GameLift::Fleet sumber daya untuk menyebarkan server game multipemain. AWS	2010-05-15
Sumber daya baru	26 Februari 2016	CloudFormation menambahkan sumber daya berikut: AWS::ECR::Repository Buat repositori Amazon Elastic Container Registry tempat pengguna dapat mendorong dan menarik gambar Docker. AWS::EC2::NatGateway Gunakan gateway penerjemah alamat jaringan (NAT) untuk mengaktifkan EC2 instance di subnet pribadi untuk terhubung ke Internet. AWS::Elasticsearch::Domain Buat domain OpenSearch Layanan Amazon yang menjalankan kluster OSS Elasticsearch lama. AWS::EMR::Cluster, AWS::EMR::InstanceGroupConfig, AWS::EMR::Step Gunakan sumber daya Amazon EMR untuk membantu Anda menganalisis dan memproses data dalam jumlah besar. Anda dapat membuat kluster dan kemudian menjalankan tugas padanya.	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Sumber daya yang diperbarui	26 Februari 2016	CloudFormation memperbarui sumber daya berikut: AWS::CloudTrail::Trail Gunakan <code>IsMultiRegionTrail</code> properti untuk menentukan apakah akan membuat AWS CloudTrail jejak di wilayah tempat Anda membuat tumpukan atau di semua wilayah. AWS::Config::ConfigurationRecorder Untuk grup catatan, gunakan properti <code>IncludeGlobalResourceTypes</code> untuk mencatat semua tipe sumber daya global. AWS::RDS::DBCluster Gunakan properti <code>KmsKeyId</code> dan <code>StorageEncrypted</code> untuk mengenkripsi instans basis data dalam klaster.	2010-05-15
Pertahankan sumber daya	26 Februari 2016	Untuk tumpukan dalam status <code>DELETE_FAILED</code> , gunakan parameter <code>RetainResources</code> untuk mempertahankan sumber daya yang tidak dapat dihapus CloudFormation . Untuk informasi lebih lanjut, lihat Menghapus Tumpukan Gagal.	2010-05-15
Perbarui tanda tumpukan	26 Februari 2016	Anda dapat menambahkan, memodifikasi, atau menghapus tanda tumpukan ketika Anda memperbarui tumpukan. Untuk informasi selengkapnya, lihat Pembaruan CloudFormation Tumpukan.	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Lanjutkan roll back gagal memperbarui rollback	25 Januari 2016	Untuk tumpukan dalam status UPDATE_ROLLBACK_FAILED , Anda dapat terus roll back pembaruan untuk mendapatkan tumpukan Anda dalam status kerja. Dengan begitu, Anda dapat mengembalikan tumpukan ke pengaturan aslinya dan mencoba memperbaruinya lagi. Untuk informasi selengkapnya, lihat Lanjutkan Roll Back Pembaruan .	2010-05-15
Templat sampel baru tersedia untuk wilayah Asia Pacific (Seoul).	7 Januari 2016	Kumpulan CloudFormation contoh template berikut adalah untuk wilayah ap-northeast-2: • Solusi Sampel • Kerangka Kerja Aplikasi • Layanan Untuk informasi selengkapnya, lihat Bekerja dengan CloudFormation templat.	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Sumber daya baru	28 Desember 2015	CloudFormation menambahkan sumber daya berikut: AWS::DirectoryService::MicrosoftAD Gunakan sumber daya Microsoft Active Directory untuk membuat direktori Microsoft Active Directory di AWS. AWS::Logs::Destination dan AWS::Logs::LogStream Gunakan sumber daya Amazon CloudWatch Logs untuk membuat tujuan pemrosesan data log secara real-time atau untuk membuat aliran log, masing-masing. AWS::WAF::ByteMatchSet, AWS::WAF::IPSet, AWS::WAF::Rule, AWS::WAF::SqlInjectionMatchSet, dan AWS::WAF::WebACL Gunakan AWS WAF sumber daya untuk mengontrol dan memantau permintaan web ke konten Anda.	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Pembaruan sumber daya	28 Desember 2015	CloudFormation memperbarui sumber daya berikut: AWS::CloudFront::Distribution Untuk konfigurasi distribusi, gunakan <code>WebACLId</code> properti untuk mengaitkan daftar kontrol akses AWS WAF web (ACL) dengan CloudFront distribusi di Amazon. Untuk perilaku cache dan perilaku cache default, Anda dapat menentukan nilai waktu untuk tayang (TTL) default dan maksimum. AWS::DynamoDB::Table Anda dapat membuat, memperbarui, atau menghapus indeks sekunder global tanpa mengganti tabel Amazon DynamoDB Anda. AWS::S3::Bucket Gunakan properti <code>ReplicationConfiguration</code> untuk menentukan objek mana yang akan direplikasi dan di mana objek tersebut disimpan. Gunakan properti <code>NotificationConfiguration</code> properti untuk menentukan filter sehingga Amazon Simple Storage Service mengirimkan pemberitahuan untuk objek yang Anda tentukan.	2010-05-15
Pengelompokan dan pengurutan parameter	3 Desember 2015	Gunakan kunci AWS::CloudFormation::Interface metadata untuk mengelompokkan dan mengurutkan parameter di CloudFormation konsol saat pengguna membuat atau memperbarui tumpukan dengan templat Anda.	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Atribut kebijakan pembaruan	3 Desember 2015	Untuk atribut kebijakan pembaruan Auto Scaling, gunakan <code>MinSuccessfulInstancesPercent</code> properti untuk menentukan persentase instance yang harus memberi sinyal keberhasilan untuk pembaruan yang berhasil.	2010-05-15
Sumber daya baru	3 Desember 2015	CloudFormation menambahkan sumber daya berikut: AWS::CodePipeline::Pipeline dan AWS::CodePipeline::CustomActionType Gunakan CodePipeline sumber daya untuk membuat pipeline yang menjelaskan bagaimana perubahan perangkat lunak melalui proses rilis. AWS::Config::ConfigurationRecorder, AWS::Config::DeliveryChannel, dan AWS::Config::ConfigRule Gunakan AWS Config sumber daya untuk memantau perubahan konfigurasi ke AWS sumber daya tertentu. AWS::KMS::Key Gunakan sumber daya AWS Key Management Service (AWS KMS) untuk membuat kunci terkelola pelanggan AWS KMS yang dapat digunakan pengguna untuk mengenkripsi sejumlah kecil data. AWS::SSM::Document Gunakan Amazon EC2 Systems Manager untuk membuat dokumen yang menentukan konfigurasi on-instance.	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Pembaruan sumber daya	3 Desember 2015	CloudFormation memperbarui sumber daya berikut: AWS::AutoScaling::LaunchConfiguration Tentukan apakah volume EBS dienkripsi. AWS::AutoScaling::ScalingPolicy Anda dapat menggunakan dua jenis kebijakan yang berbeda (penskalaan sederhana dan langkah) untuk menentukan skala grup Auto Scaling saat alarm CloudWatch Amazon CloudWatch () dilanggar. AWS::CloudTrail::Trail Gunakan CloudWatch properti untuk mengirim log ke grup CloudWatch log. Anda dapat menambahkan tag ke jejak dan menentukan AWS KMS key yang ingin Anda gunakan untuk mengenkripsi log. AWS::CodeDeploy::Application, AWS::CodeDeploy::DeploymentConfig, dan AWS::CodeDeploy::DeploymentGroup Gunakan <code>ApplicationName</code>, <code>DeploymentConfigName</code>, dan <code>DeploymentGroupName</code> properti untuk menentukan nama kustom untuk CodeDeploy sumber daya. AWS::DynamoDB::Table Gunakan <code>StreamSpecification</code> properti untuk menentukan pengaturan untuk menangkap perubahan pada item yang disimpan dalam tabel Amazon DynamoDB (DynamoDB).	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
		AWS::EC2::Instance Gunakan SsmAssociations properti untuk mengaitkan dokumen Amazon EC2 Systems Manager dengan instance. AWS::EC2::SpotFleet Gunakan properti AllocationStrategy untuk menentukan bagaimana mengalokasikan kapasitas target di seluruh kolam Spot. Gunakan properti ExcessCapacityTerminationPolicy untuk menentukan bagaimana instans diakhiri jika kapasitas target di bawah ukuran armada Spot. AWS::Redshift::Cluster Gunakan KmsKeyId properti untuk menentukan data AWS KMS key untuk mengenkripsi dalam kluster Amazon Redshift. AWS::WorkSpaces::Workspace Gunakan properti enkripsi untuk mngenkripsi data yang disimpan pada volume.	
Pembaruan sumber daya	04 Nopember 2015	Untuk AWS::EC2::Volume sumber daya, gunakan AutoEnableIO properti untuk melanjutkan I/O operasi secara otomatis jika data volume menjadi tidak konsisten .	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Sumber daya baru	1 Oktober 2015	CloudFormation menambahkan sumber daya berikut: AWS::CodeDeploy::Application, AWS::CodeDeploy::DeploymentGroup, dan AWS::CodeDeploy::DeploymentConfig Gunakan CodeDeploy sumber daya untuk membuat dan menerapkan penerapan ke EC2 atau instance lokal. AWS::DirectoryService::SimpleAD Gunakan sumber daya Direktori Aktif Sederhana untuk membuat Simple AD AWS Directory Service , yang merupakan direktori Direktori Aktif Microsoft yang kompatibel. AWS::EC2::PlacementGroup Gunakan grup penempatan untuk membuat kluster instans di jaringan latensi rendah. AWS::EC2::SpotFleet Gunakan armada Spot untuk meluncurkan koleksi instans Spot yang menjalankan tugas yang tidak dapat diinterupsi. AWS::Lambda::EventSourceMapping Gunakan sumber daya pemetaan sumber peristiwa untuk menentukan aliran sebagai sumber peristiwa untuk fungsi AWS Lambda (Lambda). AWS::Lambda::Permission Gunakan izin Lambda untuk menambahkan pernyataan ke kebijakan fungsi Lambda.	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
		AWS::Logs::SubscriptionFilter Gunakan filter langganan untuk menentukan peristiwa log mana yang dikirimkan ke aliran Kinesis Anda. AWS::RDS::DBCluster dan AWS::RDS::DBClusterParameterGroup Gunakan sumber daya grup parameter klaster dan klaster untuk membuat klaster Amazon Aurora DB. AWS::WorkSpaces::Workspace Gunakan WorkSpaces untuk membuat pengalaman desktop berbasis cloud.	
Pembaruan sumber daya	1 Oktober 2015	CloudFormation memperbarui sumber daya berikut: AWS::ElastiCache::ReplicationGroup Gunakan fungsi intrinsik <code>Fn::GetAtt</code> untuk mendapatkan daftar alamat dan port replika hanya baca. AWS::OpsWorks::Stack Gunakan <code>AgentVersion</code> properti untuk menentukan AWS OpsWorks agen tertentu. AWS::OpsWorks::App Gunakan <code>Environment</code> properti untuk menentukan variabel lingkungan untuk AWS OpsWorks aplikasi. AWS::S3::Bucket Untuk NotificationConfiguration properti, Anda dapat mengonfigurasi pengaturan notifikasi untuk fungsi Lambda dan antrian Amazon Simple Queue Service (Amazon SQS).	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Kunci kondisi IAM	1 Oktober 2015	Untuk kebijakan AWS Identity and Access Management (IAM), gunakan kunci kondisi CloudFormation khusus untuk menentukan kapan kebijakan IAM diterapkan. Untuk informasi selengkapnya, lihat Mengontrol Akses dengan AWS Identity and Access Management .	2010-05-15
AWS CloudFormation Desainer	1 Oktober 2015	Gunakan AWS CloudFormation Designer untuk membuat dan memodifikasi template menggunakan drag-and-drop antarmuka.	2010-05-15
Sumber daya baru	24 Agustus 2015	Gunakan AWS::EC2::VPCEndpoint sumber daya untuk membuat koneksi pribadi antara VPC Anda dan layanan lain AWS .	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Pembaruan sumber daya	24 Agustus 2015	CloudFormation memperbarui sumber daya berikut: AWS::ElasticBeanstalk::Environment Gunakan Tags properti untuk menentukan tag (pasangan kunci-nilai) untuk lingkungan AWS Elastic Beanstalk (Elastic Beanstalk). AWS::Lambda::Function Untuk Code properti, gunakan ZipFile properti untuk menulis kode sumber fungsi Lambda Anda secara langsung di template. Saat ini, Anda dapat menggunakan properti ZipFile hanya untuk lingkungan waktu aktif nodejs. Anda masih dapat menunjuk ke file dalam bucket S3 untuk semua lingkungan waktu aktif, seperti java8 dan nodejs. AWS::OpsWorks::Instance Gunakan EbsOptimized properti untuk menunjukkan apakah instance dioptimalkan untuk Amazon Elastic Block Store (Amazon EBS) I/O. AWS::RDS::DBInstance Untuk properti SourceDBInstanceIdentifier , Anda dapat menentukan instans basis data di wilayah lain untuk membuat replika baca lintas-wilayah.	2010-05-15
URL templat Amazon S3	24 Agustus 2015	Untuk bucket berkemampuan versi, Anda dapat menentukan ID versi di URL templat Amazon S3 saat membuat atau memperbarui tumpukan, seperti. https://s3.amazonaws.com/templates/myTemplate.template?versionId=123ab1cdeKd0W5IH4GAcYbEngcpTJTJTDW	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Sumber daya baru	3 Agustus 2015	Gunakan AWS::EFS::FileSystem sumber daya untuk membuat sistem file Amazon Elastic File System (Amazon EFS) dan AWS::EFS::MountTarget sumber daya untuk membuat titik pemasangan untuk sistem file.	2010-05-15
Perubahan persyaratan izin	11 Juni 2015	Saat Anda membuat atau memperbarui AWS::RDS::DBInstances sumber daya, Anda sekarang juga harus memiliki izin untuk memanggil <code>ec2:DescribeAccountAttributes</code> tindakan.	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Sumber daya baru	11 Juni 2015	CloudFormation menambahkan sumber daya berikut: AWS::DataPipeline::Pipeline Gunakan alur data untuk mengotomatisasi gerakan dan transformasi data. Sumber daya Amazon Elastic Container Service Gunakan AWS::ECS::Service, AWS::ECS::Cluster, dan AWS::ECS::TaskDefinition sumber daya untuk membuat kontainer Docker pada sekelompok EC2 instance. AWS::ElastiCache::ReplicationGroup Gunakan grup replikasi untuk membuat koleksi simpul dengan satu klaster baca-tulis primer dan maksimum lima klaster hanya baca sekunder. AWS::IAM::ManagedPolicy Gunakan kebijakan terkelola untuk membuat kebijakan di AWS akun yang dapat Anda gunakan untuk menerapkan izin kepada pengguna, grup, dan peran IAM. AWS::Lambda::Function Gunakan fungsi Lambda untuk menjalankan kode sebagai respons terhadap peristiwa. AWS::RDS::OptionGroup Gunakan grup opsi untuk membantu Anda membuat dan mengelola database Amazon Relational Database Service (Amazon RDS).	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Pembaruan sumber daya	11 Juni 2015	CloudFormation memperbarui sumber daya berikut: AWS::EC2::Subnet Gunakan properti <code>MapPublicIpOnLaunch</code> untuk menetapkan alamat IP publik secara otomatis untuk instans di subnet. AWS::ElastiCache::CacheCluster Gunakan properti <code>SnapshotName</code> untuk mengembalikan data snapshot ke dalam klaster cache Redis baru. AWS::IAM::User Untuk properti <code>LoginProfile</code>, gunakan properti <code>PasswordResetRequired</code> sehingga pengguna diperlukan untuk mengatur kata sandi baru ketika mereka masuk ke AWS Management Console. AWS::OpsWorks::Layer Gunakan <code>LifecycleEventConfiguration</code> properti untuk mengonfigurasi peristiwa siklus hidup untuk lapisan. AWS OpsWorks AWS::S3::Bucket Untuk properti <code>LifecycleConfiguration</code>, gunakan properti <code>NoncurrentVersionExpirationInDays</code> dan <code>NoncurrentVersionTransition</code> untuk menentukan aturan siklus hidup untuk versi objek tidak saat ini.	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Tipe parameter baru	19 Mei 2015	Setiap kali Anda menggunakan CloudFormation konsol untuk membuat atau memperbarui tumpukan, Anda dapat mencari nilai tipe parameter AWS-specific berdasarkan ID, nama, atau nilai tag Nama. CloudFormation juga menambahkan dukungan untuk jenis parameter AWS-spesifik berikut. Untuk informasi lebih lanjut, lihat 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>Daftar< > AWS::EC2::Image::Id</code> • <code>AWS::EC2::SecurityGroup::GroupName</code> • <code>Daftar< > AWS::EC2::SecurityGroup::GroupName</code> • <code>AWS::EC2::Volume::Id</code> • <code>Daftar< > AWS::EC2::Volume::Id</code> • <code>AWS::Route53::HostedZone::Id</code> • <code>List<AWS::Route53::HostedZone::Id></code>	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Sumber daya baru	16 April 2015	CloudFormation menambahkan sumber daya berikut: AWS::AutoScaling::LifecycleHook Gunakan kait siklus hidup Auto Scaling untuk mengontrol status instance setelah diluncurkan atau dihentikan. AWS::RDS::EventSubscription Gunakan langganan acara untuk mendapatkan pemberitahuan tentang acara Amazon RDS.	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Pembaruan sumber daya	16 April 2015	CloudFormation memperbarui sumber daya berikut: AWS::AutoScaling::AutoScalingGroup Gunakan properti <code>NotificationConfigurations</code> untuk menentukan beberapa notifikasi. AWS::AutoScaling::LaunchConfiguration Gunakan properti <code>PlacementTenancy</code> untuk menentukan penghunian instans. Gunakan <code>ClassicLinkVPCSecurityGroups</code> properti <code>ClassicLinkVPCId</code> dan untuk menautkan instance EC2 -Classic ke VPC yang ClassicLink diaktifkan. AWS::AutoScaling::ScalingPolicy Gunakan properti <code>MinAdjustmentStep</code> untuk menentukan jumlah minimum instans yang ditambahkan atau dihapus selama acara penskalaan. AWS::CloudFront::Distribution Untuk sertifikat penampil, gunakan properti <code>MinimumProtocolVersion</code> untuk menentukan versi protokol minimum. Untuk perilaku cache, gunakan <code>CachedMethods</code> properti untuk menentukan metode mana Amazon CloudFront (CloudFront) menyimpan respons untuk cache. Untuk asal, gunakan <code>OriginPath</code> untuk menentukan jalur yang CloudFront digunakan untuk meminta konten. AWS::ElastiCache::CacheCluster Untuk cluster cache Memcached, gunakan <code>PreferredAvailabilityZones</code> properti	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
		AZMode and untuk menentukan node di beberapa Availability Zones (). AZs AWS::EC2::Volume Gunakan KmsKeyId properti untuk menentukan kunci terkelola pelanggan untuk volume terenkripsi. AWS::OpsWorks::Instance Gunakan properti TimeBasedAutoScaling untuk menskalakan instans secara otomatis berdasarkan jadwal yang Anda tentukan. AWS::OpsWorks::Layer Gunakan properti LoadBasedAutoScaling untuk menentukan kebijakan penskalaan berbasis beban. Untuk konfigurasi volume, gunakan Iops properti VolumeType and untuk menentukan jenis volume dan jumlah I/O operasi per detik, masing-masing. AWS::RDS::DBInstance Gunakan properti CharacterSetName untuk menentukan set karakter untuk mesin basis data yang didukung. Gunakan StorageEncrypted properti untuk menunjukkan apakah instance database akan dienkripsi dan KmsKeyId untuk menentukan kunci terkelola pelanggan untuk instance database terenkripsi. AWS::Route53::HealthCheck Gunakan properti HealthCheckTags untuk menghubungkan tanda dengan pemeriksaan kondisi.	

Perubahan	Tanggal rilis	Deskripsi	Versi API
		AWS::Route53::HostedZone Gunakan properti VPCs untuk membuat zona yang di-hosting privat. Gunakan properti <code>HostedZoneTags</code> untuk menghubungkan tanda dengan zona yang di-hosting.	
Bagian templat baru	16 April 2015	Tambahkan bagian Metadata ke templat Anda untuk memasukkan objek JSON acak yang menjelaskan templat Anda, seperti detail desain atau implementasi.	2010-05-15
Pembaruan sumber daya	8 April 2015	Untuk AWS::CloudFormation::CustomResource sumber daya, Anda dapat menentukan fungsi Lambda Amazon Resource Names (ARNs) di properti <code>ServiceToken</code>	2010-05-15
Pembaruan Amazon RDS	24 Desember 2014	CloudFormation menambahkan dua properti baru untuk instance RDS DB. Anda dapat menghubungkan grup opsi dengan instans DB dan menentukan tipe penyimpanan instans DB. Untuk informasi selengkapnya, lihat AWS::RDS::DBInstance .	2010-05-15
Pembaruan Elastic Load Balancing	24 Desember 2014	Anda dapat menggunakan properti <code>ConnectionSettings</code> untuk menentukan berapa lama koneksi dapat tetap idle. Untuk informasi selengkapnya, lihat AWS::ElasticLoadBalancing::LoadBalancer .	2010-05-15
Pembaruan Route 53	6 November 2014	Anda sekarang dapat menyediakan dan mengelola zona yang dihosting Route 53, pemeriksaan kesehatan , kumpulan catatan failover , dan kumpulan catatan geolokasi .	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Peningkatan pembaruan bergulir Auto Scaling	6 November 2014	Selama pembaruan, Anda dapat menggunakan <code>WaitOnResourceSignals</code> bendera untuk menginstruksikan CloudFormation agar menunggu instance memberi sinyal keberhasilan. Dengan cara itu, CloudFormation tidak akan memperbarui batch instans berikutnya sampai batch saat ini siap. Untuk informasi selengkapnya, lihat UpdatePolicyAtribut .	2010-05-15
VPC Fn baru: atribut GetAtt	6 November 2014	Dengan ID VPC, Anda dapat mengambil grup keamanan default dan ACL jaringan untuk VPC tersebut. Untuk informasi selengkapnya, lihat Fn::GetAtt .	2010-05-15
Jenis parameter AWS spesifik baru	6 November 2014	Anda dapat menentukan jenis parameter AWS-spesifik di CloudFormation template Anda. Di CloudFormation konsol, tipe parameter ini menyediakan daftar drop-down dari nilai yang valid. Dengan API atau AWS CLI, CloudFormation dapat dengan cepat memvalidasi nilai untuk jenis parameter ini sebelum membuat atau memperbarui tumpukan. Untuk informasi lebih lanjut, lihat Parameter .	2010-05-15
CreationPolicy atribut	6 November 2014	Dengan CreationPolicy atribut, Anda dapat menginstruksikan CloudFormation untuk menunggu hingga aplikasi siap pada EC2 instance sebelum melanjutkan pembuatan tumpukan. Anda dapat menggunakan kebijakan pembuatan alih-alih syarat tunggu dan penanganan syarat tunggu. Untuk informasi selengkapnya, lihat CreationPolicyAtribut .	2010-05-15
Nilai yang CloudFront diteruskan Amazon	29 September 2014	Untuk perilaku cache, Anda dapat meneruskan header ke asal. Lihat ForwardedValues .	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
AWS OpsWorks diperbarui	29 September 2014	Untuk Chef 11.10, Anda dapat menggunakan properti <code>ChefConfiguration</code> untuk mengaktifkan Berkshelf. Anda juga dapat menggunakan grup keamanan AWS OpsWorks bawaan dengan AWS OpsWorks tumpukan Anda. Untuk informasi selengkapnya, lihat AWS::OpsWorks::Stack .	2010-05-15
Dukungan penandaan Elastic Load Balancing	29 September 2014	AWS CloudFormation tag penyeimbang beban Elastic Load Balancing dengan tag tingkat tumpukan. Anda juga dapat menambahkan tanda Anda sendiri ke penyeimbang beban. Lihat AWS::ElasticLoadBalancing::LoadBalancer .	2010-05-15
Pembaruan kebijakan topik Layanan Pemberitahuan Sederhana Amazon	29 September 2014	Anda sekarang dapat memperbarui kebijakan topik Amazon SNS. Untuk informasi selengkapnya, lihat AWS::SNS::TopicPolicy .	2010-05-15
Pembaruan instans DB RDS	5 September 2014	Anda dapat menentukan apakah instans DB menghadap Internet dengan menggunakan <code>PubliclyAccessible</code> properti di sumber daya. AWS::RDS::DBInstance	2010-05-15
UpdatePolicy pembaruan atribut	05 September 2014	Anda dapat menentukan kebijakan pembaruan untuk grup Auto Scaling yang memiliki tindakan terjadwal terkait. Untuk informasi selengkapnya, lihat UpdatePolicyAtribut .	2010-05-15
CloudWatch Dukungan Amazon	10 Juli 2014	Anda dapat menggunakannya CloudFormation untuk menyediakan dan mengelola grup CloudWatch log Amazon CloudWatch Log (Log) dan filter metrik. Untuk informasi selengkapnya, lihat AWS::Logs::LogGroup atau AWS::Logs::MetricFilter .	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Pembaruan konfigurasi CloudFront distribusi Amazon	17 Juni 2014	Anda dapat menentukan properti konfigurasi CloudFront distribusi tambahan: • Respons kesalahan kustom menetapkan pesan kesalahan kustom untuk kode status HTTP 4xx dan 5xx. • Kelas harga menetapkan harga maksimum yang ingin Anda bayar untuk layanan CloudFront . • Pembatasan menetapkan siapa yang dapat melihat konten Anda. • Sertifikat penampil menentukan sertifikat yang akan digunakan saat penampil menggunakan HTTPS. • Untuk perilaku cache, Anda dapat menentukan metode HTTP yang diizinkan dan menunjukkan apakah akan meneruskan cookie. Untuk informasi selengkapnya, lihat AWS::CloudFront::Distribution.	2010-05-15
EC2 pembaruan contoh	17 Juni 2014	Anda dapat menentukan apakah instans berhenti atau berakhir ketika Anda memanggil perintah penonaktifan sistem operasi instans. Untuk informasi selengkapnya, lihat AWS::EC2::Instance.	2010-05-15
Pembaruan volume EBS	17 Juni 2014	Anda dapat menggunakan volume EBS terenkripsi dengan tipe instans yang didukung. Untuk informasi selengkapnya, lihat AWS::EC2::Volume.	2010-05-15
Koneksi peering VPC Amazon baru	17 Juni 2014	Anda dapat menggunakan CloudFormation untuk membuat koneksi peering Amazon Virtual Private Cloud (Amazon VPC), yang membuat koneksi jaringan antara keduanya. VPCs Untuk informasi selengkapnya, lihat AWS::EC2::VPCPeeringConnection.	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Pembaruan grup EC2 Auto Scaling Amazon	17 Juni 2014	Anda dapat menentukan grup penempatan klaster yang ada untuk meluncurkan instance untuk grup EC2 Auto Scaling Amazon. Untuk informasi selengkapnya, lihat AWS::AutoScaling::AutoScalingGroup .	2010-05-15
AWS CloudTrail dukungan	17 Juni 2014	CloudFormation support AWS CloudTrail, yang dapat menangkap panggilan API yang dibuat dari AWS akun Anda dan mempublikasikan log di lokasi yang Anda tentukan. Untuk informasi selengkapnya, lihat AWS::CloudTrail::Trail .	2010-05-15
Perbarui peningkatan tumpukan	12 Mei 2014	CloudFormation mendukung fitur tambahan untuk memperbarui tumpukan: • Anda dapat memperbarui parameter CloudFormation tumpukan tanpa mengirimkan ulang template tumpukan. • Anda dapat menambah atau menghapus topik notifikasi Amazon SNS untuk tumpukan. CloudFormation Untuk informasi selengkapnya, lihat Pembaruan CloudFormation Tumpukan .	2010-05-15
Dukungan Amazon Kinesis	6 Mei 2014	Anda dapat menggunakan CloudFormation untuk membuat aliran Amazon Kinesis yang menangkap dan mengangkut catatan data dari sumber data. Untuk informasi selengkapnya, lihat AWS::Kinesis::Stream .	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Properti bucket S3 baru	5 Mei 2014	CloudFormation mendukung properti bucket S3 tambahan: • Berbagi sumber daya lintas-asal (CORS) menetapkan pembagian sumber daya lintas-asal pada objek dalam bucket. • Siklus hidup menentukan cara Amazon S3 mengelola objek selama masa pakainya. • Kebijakan pencatatan akses merekam informasi tentang permintaan yang dibuat ke bucket Anda. • Pemberitahuan menentukan acara mana yang akan dilaporkan dan topik Amazon SNS mana yang akan dikirimkan pesan. • Versioning mengaktifkan beberapa varian dari semua objek dalam bucket. • Aturan pengalihan dan perutean mengatur perilaku pengalihan untuk permintaan yang dibuat ke titik akhir situs web bucket. Untuk informasi selengkapnya, lihat AWS::S3::Bucket.	2010-05-15
Dukungan EC2 Auto Scaling Amazon	5 Mei 2014	CloudFormation mendukung pengumpulan metrik untuk grup Auto Scaling. Untuk informasi selengkapnya, lihat AWS::AutoScaling::AutoScalingGroup.	2010-05-15
Pembaruan Fn::If	5 Mei 2014	Anda dapat menggunakan fungsi intrinsik Fn::If di bagian output templat. Untuk informasi selengkapnya, lihat Fungsi Syarat.	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Pencatatan API dengan AWS CloudTrail	2 April 2014	Anda dapat menggunakan AWS CloudTrail (CloudTrail) untuk mencatat CloudFormation permintaan. Dengan CloudTrail Anda bisa mendapatkan riwayat panggilan CloudFormation API untuk akun Anda. Untuk informasi selengkapnya, lihat Mencatat Panggilan CloudFormation API dengan AWS CloudTrail .	2010-05-15
Pembaruan Elastic Load Balancing	20 Maret 2014	Anda dapat menentukan kebijakan pencatatan akses untuk merekam informasi tentang permintaan yang dibuat ke penyeimbang beban Anda. Anda juga dapat menentukan kebijakan connection draining yang menjelaskan cara menangani permintaan yang sedang berjalan ketika instans dibatalkan pendaftarannya atau menjadi tidak sehat. Untuk informasi selengkapnya, lihat AWS::ElasticLoadBalancing::LoadBalancer .	2010-05-15
AWS OpsWorks dukungan	3 Maret 2014	Anda dapat menggunakan CloudFormation untuk menyediakan dan mengelola AWS OpsWorks tumpukan. Untuk informasi selengkapnya, lihat AWS::OpsWorks::Stack atau Cuplikan AWS OpsWorks Templat .	2010-05-15
Batas ukuran template Amazon S3 meningkat	18 Februari 2014	Anda dapat menentukan ukuran template hingga 460.800 byte di Amazon S3.	2010-05-15
Dukungan Amazon Redshift	10 Februari 2014	Anda dapat menggunakan CloudFormation untuk menyediakan dan mengelola kluster Amazon Redshift. Untuk informasi selengkapnya, lihat Cuplikan Templat Amazon Redshift atau AWS::Redshift::Cluster .	2010-05-15
Pembaruan kebijakan bucket S3 dan bucket	10 Februari 2014	Anda dapat memperbarui beberapa properti bucket S3 dan sumber daya kebijakan bucket. Untuk informasi selengkapnya, lihat AWS::S3::Bucket atau AWS::S3::BucketPolicy .	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Pembaruan lingkungan Elastic Beanstalk dan versi aplikasi	10 Februari 2014	Anda dapat memperbarui konfigurasi lingkungan Elastic Beanstalk dan versi aplikasi. Untuk informasi selengkapnya, lihat AWS::ElasticBeanstalk::Environment , AWS::ElasticBeanstalk::ConfigurationTemplate , atau AWS::ElasticBeanstalk::ApplicationVersion .	2010-05-15
Pembaruan Amazon SQS	29 Januari 2014	Anda dapat menentukan antrian huruf mati untuk antrian Amazon SQS. Untuk informasi selengkapnya, lihat AWS::SQS::Queue .	2010-05-15
Tindakan terjadwal Auto Scaling	27 Januari 2014	Anda dapat EC2 menskalakan jumlah instance dalam grup Auto Scaling berdasarkan jadwal. Dengan menggunakan jadwal, Anda dapat menskalakan aplikasi sebagai respons terhadap perubahan beban yang dapat diprediksi. Untuk informasi selengkapnya, lihat AWS::AutoScaling::ScheduledAction .	2010-05-15
Indeks sekunder DynamoDB	27 Januari 2014	Anda dapat membuat indeks sekunder lokal dan global untuk database DynamoDB. Dengan menggunakan indeks sekunder, Anda dapat mengakses data secara efisien dengan atribut selain kunci primer. Untuk informasi selengkapnya, lihat AWS::DynamoDB::Table .	2010-05-15
Pembaruan Auto Scaling	2 Januari 2014	Anda dapat menentukan ID instans untuk grup Auto Scaling atau konfigurasi peluncuran. Anda juga dapat menentukan properti perangkat blok Auto Scaling tambahan. Untuk informasi selengkapnya, lihat AWS::AutoScaling::AutoScalingGroup atau AWS::AutoScaling::LaunchConfiguration .	2010-05-15
Pembaruan Amazon SQS	2 Januari 2014	Anda dapat memperbarui antrean SQS dan menentukan properti tambahan. Untuk informasi selengkapnya, lihat AWS::SQS::Queue .	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Batas meningkat	2 Januari 2014	Anda dapat menentukan hingga 60 parameter dan 60 output di templat CloudFormation Anda.	2010-05-15
Konsol baru	19 Desember 2013	CloudFormation Konsol baru menambahkan fitur seperti peristiwa tumpukan penyegaran otomatis dan urutan abjad parameter tumpukan.	2010-05-15
Penyeimbangan beban lintas zona	19 Desember 2013	Dengan penyeimbangan beban lintas zona, Anda dapat merutekan lalu lintas ke instans back-end di semua Availability Zones (). AZs Untuk informasi selengkapnya, lihat AWS::ElasticLoadBalancing::LoadBalancer .	2010-05-15
AWS Elastic Beanstalk tingkatan lingkungan	19 Desember 2013	Anda dapat menentukan apakah AWS Elastic Beanstalk menyediakan sumber daya untuk mendukung server web atau untuk menangani tugas pemrosesan latar belakang. Untuk informasi selengkapnya, lihat AWS::ElasticBeanstalk::Environment .	2010-05-15
Nama sumber daya	19 Desember 2013	Anda dapat menetapkan nama (fisik IDs) ke sumber daya berikut: • ElastiCache kluster • Penyeimbang beban Elastic Load Balancing • Instans DB RDS Untuk informasi selengkapnya, lihat Tipe Nama.	2010-05-15
Dukungan VPC	22 November 2013	Anda dapat mengaktifkan gateway privat virtual (VGW) untuk menyebarkan rute ke tabel rute VPC. Untuk informasi selengkapnya, lihat AWS::EC2::VPNGatewayRoutePropagation .	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Buat sumber daya secara kondisional dan tetapkan properti	8 November 2013	Dengan menggunakan parameter input, Anda dapat mengendalikan pembuatan dan pengaturan sumber daya tumpukan yang ditunjuk dengan menetapkan syarat di templat CloudFormation Anda. Misalnya, Anda dapat menggunakan syarat untuk membuat sumber daya tumpukan untuk lingkungan produksi. Dengan menggunakan templat yang sama, Anda dapat membuat sumber daya tumpukan yang sama dengan kapasitas yang lebih rendah untuk lingkungan uji. Untuk informasi selengkapnya, lihat Fungsi Syarat .	2010-05-15
Cegah pembaruan yang tidak disengaja untuk menumpuk sumber daya	8 November 2013	Anda dapat mencegah pembaruan tumpukan yang mungkin mengakibatkan perubahan yang tidak disengaja untuk menumpuk sumber daya. Misalnya, jika Anda memiliki tumpukan dengan lapisan basis data yang jarang diperbarui, Anda bisa mengatur kebijakan tumpukan yang mencegah sebagian besar pengguna dari memperbarui lapisan basis data tersebut. Untuk informasi selengkapnya, lihat Mencegah Pembaruan untuk Sumber Daya Tumpukan .	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Memberi nama sumber daya	8 November 2013	Alih-alih menggunakan fisik CloudFormation yang dihasilkan IDs, Anda dapat menetapkan nama ke sumber daya tertentu. CloudFormation Sumber daya berikut mendukung penamaan • CloudWatch Alarm Amazon • Tabel DynamoDB • AWS Elastic Beanstalk aplikasi dan lingkungan • Bucket Amazon S3 • Topik Amazon SNS • Antrean Amazon SQS Untuk informasi selengkapnya, lihat Tipe Nama.	2010-05-15
Tetapkan tipe sumber daya kustom	8 November 2013	Dalam templat Anda, Anda dapat menentukan tipe sumber daya Anda sendiri untuk sumber daya kustom CloudFormation (<code>AWS::CloudFormation::CustomResource</code>). Dengan menggunakan nama tipe sumber daya kustom Anda sendiri, Anda dapat dengan cepat mengidentifikasi tipe sumber daya kustom yang Anda miliki di tumpukan Anda. Misalnya, Anda dapat menentukan <code>"Type": "Custom:: MyCustomResource"</code>. Untuk informasi selengkapnya, lihat AWS::CloudFormation::CustomResource.	2010-05-15
Tambahkan parameter semu	8 November 2013	Anda sekarang dapat merujuk ke AWS accountID CloudFormation di dalam template dengan mengacu <code>AWS::AccountId</code> pada parameter pseudo. Untuk informasi selengkapnya, lihat Referensi Parameter Semu.	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Tentukan tumpukan dalam kebijakan IAM	8 November 2013	Anda dapat mengizinkan atau menolak pengguna, grup, atau peran IAM untuk beroperasi pada CloudFormation tumpukan tertentu. Misalnya, Anda dapat menolak tindakan hapus tumpukan pada ID tumpukan tertentu. Untuk informasi selengkapnya, lihat Mengontrol Akses dengan AWS Identity and Access Management .	2010-05-15
Dukungan Federasi	14 Oktober 2013	CloudFormation mendukung kredensial keamanan sementara dari peran IAM, yang memungkinkan skenario seperti federasi dan sistem masuk tunggal ke AWS Management Console. Anda juga dapat melakukan panggilan ke CloudFormation dari EC2 instans tanpa menyematkan kredensial keamanan jangka panjang dengan menggunakan peran IAM. Untuk informasi selengkapnya tentang CloudFormation dan IAM, lihat Mengontrol Akses dengan AWS Identity and Access Management .	2010-05-15
Amazon RDS membaca dukungan replika	24 September 2013	Anda sekarang dapat membuat replika baca Amazon RDS dari instans DB sumber. Untuk informasi selengkapnya, lihat <code>SourceDBInstanceIdentifier</code> properti di AWS::RDS::DBInstance sumber daya.	2010-05-15
Kaitkan alamat IP publik dengan instans dalam grup Auto Scaling	19 September 2013	Anda sekarang dapat mengaitkan alamat IP publik dengan instance dalam grup Auto Scaling. Untuk informasi selengkapnya, lihat AWS::AutoScaling::LaunchConfiguration .	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Dukungan VPC tambahan	17 September 2013	CloudFormation menambahkan beberapa penyempurnaan untuk mendukung fungsionalitas VPC dan VPN • Anda dapat mengaitkan alamat IP publik dan beberapa alamat IP pribadi ke antarmuka EC2 jaringan Amazon. Untuk informasi selengkapnya, lihat AWS::EC2::NetworkInterface. Anda juga dapat menghubungkan alamat IP privat primer ke alamat IP elastis (EIP). • Anda dapat mengaktifkan dukungan DNS dan menentukan nama host DNS. Untuk informasi selengkapnya, lihat AWS::EC2::VPC. • Anda dapat menentukan rute statis antara gateway privat virtual ke gateway VPN Anda. Untuk informasi selengkapnya, lihat AWS::EC2::VPNConnectionRoute.	2010-05-15
Dukungan grup keamanan Redis dan VPC untuk Amazon ElastiCache	3 September 2013	Anda sekarang dapat menentukan Redis sebagai mesin cache untuk klaster Amazon ElastiCache (ElastiCache). Anda juga sekarang dapat menetapkan grup ElastiCache keamanan VPC ke cluster. Untuk informasi selengkapnya, lihat AWS::ElastiCache::CacheCluster.	2010-05-15
Pembuatan tumpukan paralel, pembaruan dan penghapusan, dan pembaruan tumpukan nest	12 Agustus 2013	AWS CloudFormation sekarang membuat, memperbarui, dan menghapus sumber daya secara paralel, meningkatkan kinerja operasi. Jika Anda memperbarui templat tingkat atas, AWS CloudFormation secara otomatis memperbarui tumpukan nest yang telah berubah. Untuk informasi selengkapnya, lihat Pembaruan CloudFormation Tumpukan.	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Grup keamanan VPC sekarang dapat diatur dalam instans DB RDS	28 Februari 2013	Anda sekarang dapat menetapkan grup keamanan VPC untuk instans DB RDS dengan CloudFormation. Untuk informasi selengkapnya, lihat properti VPCSecurityGrup di AWS::RDS::DBInstance .	2010-05-15
Penerapan bergulir untuk grup Auto EC2 Scaling Amazon	20 Februari 2013	CloudFormation sekarang mendukung kebijakan pembaruan di grup EC2 Auto Scaling Amazon, yang menjelaskan cara instance di grup EC2 Auto Scaling Amazon diganti atau dimodifikasi saat grup Amazon Auto Scaling EC2 menambahkan atau menghapus instance. Anda dapat memodifikasi pengaturan ini pada pembuatan tumpukan atau selama pembaruan tumpukan. Untuk informasi selengkapnya dan contoh, lihat UpdatePolicyAtribut .	2010-05-15
Tindakan pembatalan dan rollback untuk pembaruan tumpukan	20 Februari 2013	CloudFormation mendukung kemampuan untuk membatalkan pembaruan tumpukan. Tumpukan harus dalam status UPDATE_IN_PROGRESS ketika permintaan pembaruan dibuat. Informasi lebih lanjut tersedia dalam topik berikut: • Membatalkan Pembaruan Stack • cancel-update-stack • CancelUpdateStack di Referensi API CloudFormation	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Instans yang dioptimalkan EBS untuk grup Amazon Auto EC2 Scaling	20 Februari 2013	Kini Anda dapat menyediakan instans yang dioptimalkan EBS di grup Amazon EC2 Auto Scaling untuk throughput khusus ke Amazon Elastic Block Store (Amazon EBS) Elastic Block Store (Amazon EBS) dalam instans berskala otomatis. Implementasinya mirip dengan dukungan yang dirilis sebelumnya untuk EC2 instans Amazon EBS yang dioptimalkan. Untuk informasi selengkapnya, lihat <code>EbsOptimized</code> properti baru di AWS::AutoScaling::LaunchConfiguration.	2010-05-15
Dokumentasi baru	21 Desember 2012	AWS::EC2::Instance sekarang menyediakan <code>BlockDeviceMappings</code> properti untuk memungkinkan Anda mengatur pemetaan perangkat blok untuk instance Anda EC2 . Dengan perubahan ini, dua tipe baru telah ditambahkan: • BlockDeviceMapping • Perangkat Blok	2010-05-15
Dokumentasi baru	21 Desember 2012	Bagian baru telah ditambahkan untuk menjelaskan prosedur untuk membuat dan melihat tumpukan menggunakan yang baru saja didesain ulang AWS Management Console. Anda dapat menemukannya di sini: • Membuat Stack dari CloudFormation konsol • Lihat informasi tumpukan dari CloudFormation konsol	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Dokumentasi baru	15 November 2012	Informasi tentang sumber daya kustom diberikan dalam topik berikut: • Sumber Daya Kustom • AWS::CloudFormation::CustomResource • Referensi Sumber Daya Kustom	2010-05-15
Dokumentasi yang diperbarui	15 November 2012	CloudFormation sekarang mendukung menentukan I/O operasi yang disediakan per detik (IOPS) untuk instans RDS DB. Anda dapat mengatur nilai ini dari 1000—10.000 dalam peningkatan 1000 IOPS dengan menggunakan properti <code>Iops</code> baru di <code>AWS::RDS::DBInstance</code> Untuk informasi selengkapnya tentang menentukan IOPS untuk instans DB RDS, lihat Provisioned IOPS dalam Panduan Pengguna Amazon Relational Database Service.	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Dokumenta si baru dan yang diperbarui	27 Agustus 2012	Topik telah direorganisasi untuk lebih jelas memberika n informasi spesifik tentang penggunaan AWS Management Console dan menggunakan antarmuka baris CloudFormation perintah (CLI). Informasi tentang penandaan CloudFormation tumpukan telah ditambahkan, termasuk panduan baru dan topik referensi yang diperbarui: • Topik baru: Konfigurasi opsi tumpukan. • Informasi baru tentang tag dalam referensi CloudForm ation API: CreateStack, Stack, dan Tag. Informasi baru tentang bekerja dengan tumpukan Windows: • Gambar Mesin Microsoft Windows Amazon (AMIs) dan AWS CloudFormation templat • Bootstrapping tumpukan Windows AWS CloudForm ation Topik baru: Menggunakan ekspresi reguler dalam CloudFormation templat.	2010-05-1 5

Perubahan	Tanggal rilis	Deskripsi	Versi API
Fitur baru	25 April 2012	CloudFormation sekarang menyediakan dukungan penuh untuk keamanan Virtual Private Cloud (VPC) dengan Amazon. EC2 Anda sekarang dapat membuat dan mengisi seluruh VPC dengan setiap jenis sumber daya VPC (subnet, gateway, ACLs jaringan, tabel rute, dan sebagainya) menggunakan satu template. CloudFormation Dokumentasi untuk tipe sumber daya berikut telah diperbarui: • AWS::EC2::SecurityGroup • AWS::EC2::SecurityGroupIngress • AWS::EC2::SecurityGroupEgress • AWS::EC2::Instance • AWS::AutoScaling::AutoScalingGroup • AWS::EC2::EIP • AWS::EC2::EIPAssociation • AWS::ElasticLoadBalancing::LoadBalancer Tipe sumber daya baru telah ditambahkan ke dokumentasi: • AWS::EC2::VPC • AWS::EC2::InternetGateway • AWS::EC2::DHCPOptions • AWS::EC2::DHCPOptions • AWS::EC2::RouteTable • AWS::EC2::NetworkAcl • AWS::EC2::NetworkAclEntry • AWS::EC2::Subnet	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
		• AWS::EC2::VPNGateway • AWS::EC2::CustomerGateway	
Fitur baru	13 April 2012	CloudFormation sekarang memungkinkan Anda untuk menambah atau menghapus elemen dari tumpukan saat memperbaruinya. CloudFormation Pembaruan Tumpukan telah diperbarui, dan bagian baru telah ditambahkan ke panduan: Ubah Sumber Daya Tumpukan , yang menjelaskan cara menambah dan menghapus sumber daya saat memperbarui tumpukan.	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Fitur baru	2 Februari 2012	CloudFormation sekarang menyediakan dukungan untuk sumber daya di Amazon Virtual Private Cloud (Amazon VPC) yang ada. Dengan rilis ini, Anda dapat: - Luncurkan instans EC2 Khusus ke dalam VPC Amazon yang sudah ada. Untuk informasi selengkapnya, lihat AWS::EC2::Instance. - Tetapkan SourceDestCheck atribut EC2 instance yang berada di VPC Amazon yang ada. Untuk informasi selengkapnya, lihat AWS::EC2::Instance. - Buat alamat IP Elastis di VPC Amazon yang ada. Untuk informasi selengkapnya, lihat AWS::EC2::EIP. - Gunakan CloudFormation untuk membuat grup keamanan VPC Amazon dan aturan masuk/keluar di VPC yang ada. Untuk informasi selengkapnya, lihat AWS::EC2::SecurityGroup. - Kaitkan grup Auto Scaling dengan VPC Amazon yang ada dengan menyetel VPCZoneIdentifier properti sumber daya Anda. <code>AWS::AutoScaling::AutoScalingGroup</code> Untuk informasi selengkapnya, lihat AWS::AutoScaling::AutoScalingGroup. - Pasang penyeimbang beban Elastic Load Balancing ke subnet Amazon VPC dan buat grup keamanan untuk penyeimbang beban. Untuk informasi selengkapnya, lihat AWS::ElasticLoadBalancing::LoadBalancer. - Buat instans RDS DB di VPC Amazon yang ada. Untuk informasi selengkapnya, lihat AWS::RDS::DBInstance.	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Fitur baru	2 Februari 2012	Anda sekarang dapat memperbarui properti untuk sumber daya berikut di tumpukan yang ada: • AWS::EC2::SecurityGroupIngress • AWS::EC2::SecurityGroupEgress • AWS::EC2::EIPAssociation • AWS::RDS::DBSubnetGroup • AWS::RDS::DBSecurityGroup • AWS::RDS::DBSecurityGroupIngress • AWS::Route53::RecordSetGroup Untuk daftar lengkap sumber daya yang dapat diperbarui dan detail tentang apa yang harus dipertimbangkan saat memperbarui tumpukan, lihat CloudFormation Pembaruan Tumpukan.	2010-05-15
Panduan yang direstrukturisasi	2 Februari 2012	Menata ulang bagian yang ada menjadi bagian baru: Bekerja dengan CloudFormation Template dan Mengelola Tumpukan. Memindahkan Referensi Templat ke Daftar Isi tingkat atas. Memindahkan Memeriksa Biaya CloudFormation Tumpukan Anda ke bagian Memulai.	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Konten baru	2 Februari 2012	Ditambahkan bagian baru: • Panduan: Memperbarui Tumpukan adalah tutorial yang memandu proses memperbarui tumpukan LAMP. • Menerapkan Aplikasi di Amazon EC2 dengan AWS CloudFormation menjelaskan cara menggunakan skrip CloudFormation pembantu untuk menyebarkan aplikasi menggunakan metadata yang disimpan dalam template Anda. • CloudFormationReferensi Skrip Pembantu menyediakan bahan referensi untuk skrip CloudFormation pembantu (cfn-init,, cfn-signal cfn-get-metadata, dan cfn-hup).	2010-05-15
Fitur baru	26 Mei 2011	CloudFormation sekarang menyediakan <code>list-stacks</code> perintah, yang memungkinkan Anda untuk daftar tumpukan difilter oleh status tumpukan. Tumpukan yang dihapus dapat dicantumkan hingga 90 hari setelah dihapus. Untuk informasi selengkapnya, lihat Menjelaskan dan Membuat Daftar Tumpukan Anda .	2010-05-15
Fitur baru	26 Mei 2011	<code>get-template</code> Perintah <code>describe-stack-resources</code> dan sekarang memungkinkan Anda untuk mendapatkan informasi dari tumpukan yang telah dihapus selama 90 hari setelah dihapus. Untuk informasi selengkapnya, lihat Daftar Sumber Daya dan Mengambil Template .	2010-05-15
Tautan baru	1 Maret 2011	CloudFormation informasi titik akhir sekarang terletak di. Referensi Umum AWS Untuk informasi lebih lanjut, kunjungi Wilayah dan Titik Akhir di AWS Referensi Umum .	2010-05-15

Perubahan	Tanggal rilis	Deskripsi	Versi API
Rilis awal	25 Februari 2011	Rilis publik awal CloudFormation.	2010-05-15

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.