



Guía del usuario

AWS CloudFormation



Versión de API 2010-05-15

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

AWS CloudFormation: Guía del usuario

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

Las marcas comerciales y la imagen comercial de Amazon no se pueden utilizar en relación con ningún producto o servicio que no sea de Amazon de ninguna manera que pueda causar confusión entre los clientes y que menosprecie o desacredite a Amazon. Todas las demás marcas registradas que no son propiedad de Amazon son propiedad de sus respectivos propietarios, que pueden o no estar afiliados, conectados o patrocinados por Amazon.

Table of Contents

Qué es CloudFormation?	1
Simplificación de la administración de la infraestructura	1
Replicación rápida de la infraestructura	1
Control y seguimiento sencillos de los cambios en la infraestructura	2
Introducción a CloudFormation	2
Información relacionada	2
Introducción	4
Funcionamiento de CloudFormation	4
Conceptos clave	4
Funcionamiento de CloudFormation	8
Formas de empezar con CloudFormation	11
Registrarse para obtener Cuenta de AWS	12
Cómo crear una Cuenta de AWS	12
Creación de un usuario con acceso administrativo	12
Creación de su primera pila	14
Requisitos previos	15
Creación de una pila de CloudFormation con la consola	15
Supervisión de la creación de una pila	19
Prueba del servidor web	20
Solución de problemas	20
Limpieza	22
Sigüientes pasos	23
Prácticas recomendadas	24
Reduzca el bucle de retroalimentación para mejorar la velocidad de entrega	25
Organice sus pilas por ciclo de vida y titularidad	26
Utilice referencias entre pilas para devolver el valor de una salida exportada por otra pila	27
Utilice StackSets de CloudFormation para implementaciones en varias cuentas y regiones	27
Verifique las cuotas de todos los tipos de recursos	28
Reutilice plantillas para replicar pilas en varios entornos	28
Uso de módulos para reutilizar configuraciones de recursos	28
Adopción de prácticas de infraestructura como código	29
No integre credenciales en sus plantillas	29
Uso de tipos de parámetros específicos de AWS	30
Uso de limitaciones de los parámetros	30

Utilice pseudoparámetros para promover la portabilidad	31
Use <code>AWS::CloudFormation::Init</code> para implementar aplicaciones de software en las instancias de Amazon EC2	31
Utilice los scripts auxiliares más recientes	32
Validar plantillas antes de usarlas	32
Validar plantillas para la conformidad de políticas de organización	32
Uso de YAML o JSON para la creación de plantillas	33
Implemente una estrategia de etiquetado integral	34
Aproveche las macros de plantillas para realizar transformaciones avanzadas	34
Administración de todos los recursos de pila a través de CloudFormation	35
Cree conjuntos de cambios antes de actualizar las pilas	35
Utilice políticas de pila para proteger los recursos.	35
Uso de AWS CloudTrail para registrar llamadas a CloudFormation	36
Utilice revisiones de código y controles de revisión para administrar las plantillas	36
Actualice sus instancias de Amazon EC2 regularmente	36
Utilice la detección de desviaciones regularmente.	36
Configure activadores de reversión para la recuperación automática	37
Implemente estrategias efectivas de refactorización de pilas	37
Utilice CloudFormation Hooks para la administración del ciclo de vida	38
Utilice el generador IaC para crear plantillas a partir de recursos existentes.	38
Uso de AWS Infrastructure Composer para el diseño visual de plantillas	39
Considere utilizar AWS Cloud Development Kit (AWS CDK) para infraestructuras complejas	40
Utilice IAM para controlar el acceso	40
Aplicación del principio de privilegios mínimos	40
Proteja los parámetros confidenciales	41
Implementación de la política como código con AWS CloudFormation Guard	41
Trabajo con plantillas	43
Dónde se almacenan las plantillas	44
Validación de plantillas	44
Introducción a las plantillas	2
Planee usar la plantilla de referencia de CloudFormation	45
Plantillas de ejemplo	46
Formato de plantilla	46
Estructura de la plantilla	47
Comentarios	49
Especificaciones	50

Más información	50
Uso de expresiones regulares	50
Secciones de la plantilla	52
Resources	53
Parameters	62
Outputs	76
Mappings	82
Metadata	89
Rules	95
Conditions	116
Transform	127
Versión del formato	129
Description	129
Infrastructure Composer	130
¿Por qué utilizar Infrastructure Composer en el modo de consola de CloudFormation?	130
¿En qué se diferencia este modo de la consola de Infrastructure Composer?	131
Servidor de lenguaje AWS CloudFormation	131
Los IDE se integran con el servidor de lenguaje AWS CloudFormation	132
Introducción	132
Inicialización de un proyecto de CloudFormation en el IDE	139
Código abierto	140
¿Necesita ayuda?	140
Generador de IaC	140
Consideraciones	141
Permisos	142
Comandos de uso frecuente	143
Migración de una plantilla al AWS CDK	143
Inicio de un escaneo de recursos	144
Ver el resumen de análisis	148
Creación de una plantilla a partir de recursos escaneados	150
Creación de una pila a partir de recursos escaneados	154
Resolución de propiedades de solo escritura	156
Obtención de valores almacenados en otros servicios	161
Consideraciones generales	162
Obtención del valor de texto sin formato de Systems Manager	163
Obtención de una cadena segura de Systems Manager	167

Obtención de valores de Secrets Manager	170
Obtención de valores de AWS	175
Sintaxis	175
Pseudoparámetros disponibles	176
Ejemplos	177
Obtención de salidas de pila	182
Exportación de valores de salida de pilas en comparación con el uso de pilas anidadas	182
Consideraciones	183
Listado de valores de salida exportados	183
Enumeración de pilas que importan un valor de salida exportado	184
Definición de recursos existentes en tiempo de ejecución	185
Descripción general	186
Ejemplo	187
Consideraciones	190
Tipos de parámetros específicos de AWS compatibles	191
Tipos de parámetros de Systems Manager admitidos	193
Tipos de parámetros de Systems Manager no admitidos	194
Tutoriales	194
Consulta de las salidas de recursos en otra pila de CloudFormation	195
Implementación de aplicaciones en Amazon EC2	200
Actualizar una pila	210
Creación de una aplicación con escalado y equilibrio de carga	227
Interconecte con una VPC en otra cuenta	241
Realizar implementaciones azul/verde de ECS	252
Fragmentos de plantillas	276
General	277
Auto Scaling	288
Consola de facturación de AWS	321
CloudFormation	324
CloudFront	331
CloudWatch	344
Registros de CloudWatch	350
DynamoDB	384
Amazon EC2	388
Amazon ECS	426
Amazon EFS	449

Elastic Beanstalk	475
Elastic Load Balancing	479
IAM	485
AWS Lambda	502
Amazon Redshift	509
Amazon RDS	519
Route 53	528
Amazon S3	536
Amazon SNS	545
Amazon SQS	545
Amazon Timestream	546
Pilas basadas en Windows	575
Arranque de pilas de Windows	577
Uso de los tipos de recursos que brinda CloudFormation	584
Recursos personalizados	586
Macros de plantillas	632
Pilas anidadas	650
Condiciones de espera	657
Creación de configuraciones de recursos reutilizables con módulos	663
Consideraciones sobre el uso de módulos	665
Comprensión del control de versiones del módulo	665
Uso de módulos en una plantilla	667
Uso de parámetros para especificar valores de módulo	668
Recursos del módulo de referencia	671
Creación y administración de pilas	673
Interfaces para gestionar las pilas	675
Creación de una pila	676
Creación de una pila	676
Configurar las opciones la pila	679
Obtenga una vista previa de la configuración de la pila	682
Ver la información de la pila	683
Actualizar la plantilla de pila	686
Comprensión de los comportamientos de actualización de los recursos de la pila	688
Actualizar pilas con conjuntos de cambios	690
Creación de un conjunto de cambios	692
Visualización de un conjunto de cambios	698

Conjuntos de cambios con reconocimiento de desviación	707
Ejecución de un conjunto de cambios	713
Eliminación de un conjunto de cambios	715
Ejemplo de conjuntos de cambio	717
Conjuntos de cambios para pilas anidadas	729
Validación de las implementaciones de pilas	736
Cómo funciona la validación previa a la implementación	737
Consideraciones	737
Requisitos previos	738
Validación de la implementación de una pila (consola)	738
Validación de la implementación de una pila (AWS CLI)	739
Tipos de validación	741
Limitaciones de recursos	741
Actualizar pilas directamente	756
Cancelar una actualización de pila	760
Cancelación de una actualización de una pila de (consola)	760
Cancelación de una actualización de una pila (AWS CLI)	761
Eliminar una pila	761
Recursos relacionados	763
Ver las pilas eliminadas	763
Monitorizar el progreso de la pila	763
Visualización de eventos de pilas	764
Visualización de eventos de la pila por operación	773
Visualización del gráfico del cronograma de implementación de la pila	777
Comprender los eventos de creación de la pila	780
Monitorización de actualizaciones de pila	782
Continuación de la restauración de una actualización	784
Determine la causa de una falla en la pila	789
Opciones de error de pila	790
Revertir su pila ante una brecha de alarma	800
Agregar disparadores de restauración durante la creación o actualización de la pila	801
Agregar disparadores de restauración a un conjunto de cambios	802
Ver los disparadores de restauración de una pila	803
Ver desencadenadores de restauración para un conjunto de cambios	803
Detectar cambios de configuración no administrados en pilas y recursos	803
¿Qué es una desviación?	804

Códigos de estado de detección de desviaciones	805
Consideraciones a tener en cuenta cuando se detectan desviaciones	808
Detección de desviaciones en una pila de CloudFormation completa	810
Detección de desviaciones en recursos de la pila individuales	816
Resolver desviaciones con una operación de importación	820
Importar recursos AWS	830
Cómo importar recursos de AWS de forma manual	831
Cómo importar recursos de AWS automáticamente	870
Reversión de una operación de importación	872
Refactorización de pilas	877
Cómo funciona la refactorización de pilas	878
Consideraciones de la refactorización de pilas	878
Requisitos previos	879
Refactorización de las pilas (consola)	879
Refactorización de las pilas (AWS CLI)	881
Limitaciones de recursos	885
Tipo de recurso compatible	890
Uso de los enlaces de creación rápida para crear pilas de CloudFormation	1108
Formato de URL	1109
Ejemplo	1110
Creación de una pila mediante un enlace de creación rápida	1111
Ejemplos de comandos para la AWS CLI y PowerShell	1112
CancelUpdateStack	1113
ContinueUpdateRollback	1114
CreateStack	1115
Implementación	1119
DeleteStack	1120
DescribeStackEvents	1121
DescribeStackResource	1123
DescribeStackResources	1124
DescribeStacks	1128
GetTemplate	1132
ListStackResources	1133
ListStacks	1135
UpdateStack	1137
ValidateTemplate	1141

Carga de artefactos locales en un bucket de S3	1144
Administración de pilas con StackSets	1148
Conceptos de StackSets	1149
Cuentas de administrador y de destino	1150
CloudFormation StackSets	1150
Modelos de permisos para StackSets	1150
Instancias de pila	1151
Operaciones de StackSets	1152
Opciones de operaciones con StackSets	1153
Etiquetas	1155
Códigos de estado de StackSets	1156
Códigos de estado de la instancia de pila	1157
Requisitos previos	1158
Habilitación de las Regiones de AWS que están deshabilitadas de forma predeterminada	1159
Concesión de permisos autoadministrados	1160
Activación del acceso de confianza	1177
Introducción con una plantilla de ejemplo	1182
Requisitos previos	1183
Creación de un StackSet con una plantilla de ejemplo desde la consola	1183
Supervisión de la creación de un StackSet	1186
Visualización de los resultados del StackSet	1187
Actualización del StackSet	1187
Cómo agregar pilas al StackSet	1188
Limpieza	1188
Siguiendo pasos	1190
Creación de StackSets (permisos autoadministrados)	1191
Creación de un StackSet con permisos autoadministrados (consola)	1191
Creación de un StackSet con permisos autoadministrados (AWS CLI)	1193
Creación de StackSets (permisos administrados por servicios)	1195
Consideraciones	1195
Creación de un StackSet con permisos administrados por servicios (consola)	1196
Creación de un StackSet con permisos administrados por servicios (AWS CLI)	1199
Habilitar-deshabilitar la implementación automática	1203
Cómo funcionan las implementaciones automáticas	1204
Consideraciones	1204
Habilitar o deshabilitar implementaciones automáticas (consola)	1205

Habilitar o deshabilitar implementaciones automáticas (AWS CLI)	1205
Actualización de StackSets	1206
Actualización del StackSet (consola)	1206
Actualización del StackSet (AWS CLI)	1209
Adición de pilas a StackSets	1210
Cómo agregar pilas a un StackSet (consola)	1211
Cómo agregar pilas a un StackSet (AWS CLI)	1213
Anulación de parámetros en pilas	1214
Anulación de parámetros en pilas (consola)	1214
Anulación de parámetros en pilas (AWS CLI)	1217
Eliminar pilas de los StackSets	1218
Eliminación de pilas del StackSet (consola)	1218
Eliminación de pilas del StackSet (AWS CLI)	1220
Eliminación de StackSets	1222
Eliminación de un StackSet (consola)	1222
Eliminación de un StackSet (AWS CLI)	1222
Eliminar roles de servicio (opcional)	1223
Puertas de cuentas de destino	1224
Requisitos	1224
Plantillas de CloudFormation para crear funciones de Lambda	1225
Elección del modo de simultaneidad	1226
Cómo funciona cada modo de simultaneidad	1226
Elección entre Tolerancia estricta a errores y Tolerancia ligera a errores en función de la velocidad de implementación	1229
Elegir el modo de simultaneidad (consola)	1232
Elegir el modo de simultaneidad (AWS CLI)	1232
Detección de desviaciones en StackSets	1233
Cómo lleva a cabo CloudFormation la detección de desviaciones en un StackSet	1233
Detección de la desviación en un StackSet (consola)	1234
Detección de la desviación en un StackSet (AWS CLI)	1236
Detención de la detección de desviaciones en un StackSet	1243
Importación de pilas en StackSets	1243
Importación de pilas autoadministradas	1244
Importación de pilas administradas por servicios	1248
Reversión de las importaciones de pilas	1253
Prácticas recomendadas	1254

Definición de la plantilla	1254
Cómo crear o agregar pilas al StackSet	1255
Actualización de pilas en un StackSet	1255
Plantillas de de ejemplo	1255
Solución de problemas	1257
Motivos comunes para el fallo en una operación de la pila	1258
Reintento de operaciones fallidas de creación o actualización de pilas	1258
No se puede eliminar una instancia de pila a causa de un error	1259
Error en la operación de importación de pila	1260
Recuento de fallos de instancias de pila para las operaciones de StackSets	1260
Sincronización de pilas con código fuente almacenado de Git	1267
Cómo funciona la sincronización de Git	1268
Cómo funciona la sincronización de Git	1268
Comentarios sobre solicitudes de extracción	1270
Archivo de implementación de la pila	1270
Archivo de plantilla de CloudFormation	1271
Repositorio de definiciones de plantilla	1271
Requisitos previos	1272
Repositorio de Git	1272
Plantilla de CloudFormation	1272
Rol de servicio de sincronización de Git	1272
Permisos de IAM para los usuarios de la consola	1276
Creación de una pila a partir del código fuente del repositorio	1279
Creación de una pila a partir del código fuente del repositorio	1279
Actualización de la pila desde el repositorio de Git	1282
Habilitación de los comentarios sobre solicitudes de extracción	1282
Panel de estado	1283
Estado de sincronización de Git	1283
Últimos eventos de sincronización	1284
Estados de pilas admitidos	1285
Administración de extensiones a través del registro de CloudFormation	1289
Documentación relacionada	1289
Conceptos	1290
Tipos de extensiones	1290
Extensiones públicas	1291
Extensiones activadas	1291

Visualización de las extensiones disponibles y activadas	1292
Extensiones públicas	1293
rol de IAM	1294
Utilice automáticamente nuevas versiones de extensiones	1296
Utilización de alias para hacer referencia a extensiones	1297
Comandos de la AWS CLI de uso común para trabajar con extensiones públicas	1298
Activación de una extensión pública	1298
Actualización de una extensión pública	1303
Desactivación de extensiones públicas	1304
Extensiones privadas	1305
Permisos de IAM	1306
Comandos de la AWS CLI de uso común para trabajar con extensiones privadas	1306
Registro de una extensión privada	1306
Anulación del registro de extensiones privadas	1309
Edición de los datos de configuración de las extensiones	1311
Permisos necesarios para usar referencias dinámicas	1311
Edición de los datos de configuración de una extensión (consola)	1312
Edición de los datos de configuración de una extensión (AWS CLI)	1313
Registro de tipos de recursos en AWS Config	1314
Impedir que se registren propiedades confidenciales en un elemento de configuración	1315
Entrega continua	1316
Tutorial: Creación de una canalización para pilas de prueba y pilas de producción	1316
Requisitos previos	1317
Información general del tutorial	1317
Paso 1: Editar el artefacto y cargarlo en un bucket de S3	1318
Paso 2: Crear la pila de la canalización	1320
Paso 3: Visualizar la pila de WordPress	1326
Paso 4: limpie los recursos	1326
Véase también	1327
Referencia de propiedades de configuración	1328
Propiedades de configuración (consola)	1328
Propiedades de configuración (objeto JSON)	1331
Véase también	1336
CloudFormationArtefactos de	1337
Archivo de plantilla de pila	1337
Archivo de configuración de plantillas	1337

Véase también	1339
Uso de funciones de anulación de parámetros con canalizaciones de CodePipeline	1339
Fn::GetArtifactAtt	1339
Fn::GetParam	1341
Véase también	1344
Guía de referencia de plantillas	1345
Seguridad	1346
Protección de las pilas para evitar su eliminación	1347
Control de quién puede cambiar la protección de terminación de pilas	1349
Impida actualizaciones en los recursos de la pila	1349
Definición de una política de pilas	1350
Definición de una política de pilas	1351
Establecimiento de una política de pilas	1355
Actualización de recursos protegidos	1357
Modificación de una política de pilas	1360
Más políticas de pilas de ejemplo	1361
Protección de los datos	1364
Cifrado en reposo	1366
Cifrado en tránsito	1366
Privacidad del tráfico entre redes	1366
Control del acceso con IAM	1366
Definición de políticas de IAM basadas en identidad para CloudFormation	1367
Reconocimiento de recursos de IAM en plantillas de CloudFormation	1376
Administración de credenciales para aplicaciones que se ejecutan en instancias de Amazon EC2	1377
Concesión de acceso temporal (acceso federado)	1378
Ejemplos de políticas basadas en identidades	1379
AWSPolíticas de administradas por	1387
Rol de servicio CloudFormation	1390
Prevención de la sustitución confusa entre servicios	1391
Solicitudes de FAS y evaluación de permisos	1393
Registro de llamadas a la API	1396
Información de CloudFormation en CloudTrail	1396
Descripción de las entradas de los archivos de registro de CloudFormation	1397
Seguridad de la infraestructura	1401
Resiliencia	1402

Validación de conformidad	1402
Configuración y análisis de vulnerabilidades	1403
Prácticas recomendadas de seguridad	1403
Utilice IAM para controlar el acceso	1403
No integre credenciales en sus plantillas	1404
Uso de AWS CloudTrail para registrar llamadas a CloudFormation	1404
AWS PrivateLink	1404
Consideraciones para los puntos de conexión de VPC de CloudFormation	1405
Creación de un punto de conexión de VPC de interfaz para CloudFormation	1406
Creación de una política de punto de conexión de VPC para CloudFormation	1407
Monitoreo con EventBridge	1409
Información general de los eventos de sincronización de CloudFormation y Git	1410
Permisos	1412
Creación de un patrón de eventos personalizado	1412
Referencia detallada de los eventos	1413
Cambio de estado del recurso	1415
Cambio de estado de la pila	1417
Cambio de estado de detección de desviaciones	1419
Cambio de estado de StackSet	1422
Cambio de estado de la instancia de la pila de StackSet	1424
Cambio de estado de la operación de StackSet	1426
Cambio de estado de sincronización de repositorios	1428
Cambio de estado de sincronización de recursos	1431
Cuotas	1434
Disponibilidad de características	1442
StackSets y macros	1442
Solución de problemas	1443
Guía para solucionar problemas	1443
Solución de errores	1444
Errores al eliminar la pila	1445
Error de dependencia	1446
Conflictos AWS Config y AWS Systems Manager	1446
Error de análisis de parámetro al pasar una lista	1447
Permisos insuficientes de IAM	1447
Valor no válido o propiedad de recurso no admitida	1447
Cuota excedida	1448

Fallo de reversión de pilas anidadas	1448
No hay actualizaciones que realizar.	1449
El recurso no se estabiliza durante una operación de creación, actualización o eliminación de pila.	1449
El grupo de seguridad no existe en la VPC	1450
Error en la restauración de actualización	1450
La condición de espera no recibió el número necesario de señales de una instancia Amazon EC2	1452
Recurso eliminado de la pila pero no borrado	1453
Cómo contactar con el servicio de soporte	1453
Solución de problemas con Amazon Q	1455
Características	1455
Funcionamiento	1456
Precios	1456
Disponibilidad en las regiones	1456
Limitaciones	1456
Historial de revisión	1457
Actualizaciones archivadas	1487

¿Qué es CloudFormation?

CloudFormation es un servicio que le ayuda a modelar y configurar sus recursos de AWS, por lo que podrá dedicar menos tiempo a la administración de dichos recursos y más tiempo a centrarse en las aplicaciones que se ejecutan en AWS. Puede crear una plantilla que describa todos los recursos de AWS que desea (como instancias de Amazon EC2 o instancias de base de datos de Amazon RDS) y CloudFormation se encargará del aprovisionamiento y la configuración de dichos recursos. No es necesario crear y configurar individualmente los recursos de AWS ni averiguar qué depende de qué; CloudFormation se encarga de todo eso. Las siguientes situaciones demuestran cómo puede ayudarle CloudFormation.

Simplificación de la administración de la infraestructura

Para una aplicación web escalable que también incluye una base de datos backend, puede utilizar un grupo de escalado automático, un equilibrador de carga de Elastic Load Balancing y una instancia de base de datos relacional de Amazon Relational Database Service. Podría usar cada servicio individual para aprovisionar estos recursos y después de crear los recursos, tendría que configurarlos para que funcionen conjuntamente. Todas estas tareas pueden aumentar la complejidad y el tiempo incluso antes de tener en funcionamiento su aplicación.

En su lugar, puede crear o modificar una plantilla de CloudFormation existente. Una plantilla describe todos los recursos y las propiedades de estos. Cuando utiliza dicha plantilla para crear una pila de CloudFormation, CloudFormation se encarga de aprovisionar el grupo de Auto Scaling, el balanceador de carga y la base de datos. Después de que se haya creado correctamente la pila, los recursos de AWS se ponen en marcha. Es igual de fácil eliminar la pila. En este caso, se eliminan todos los recursos de la pila. Al utilizar CloudFormation, es fácil administrar un conjunto de recursos como una única unidad.

Replicación rápida de la infraestructura

Si su aplicación necesita más disponibilidad, puede replicarla en varias regiones, de modo que si una región deja de estar disponible, los usuarios pueden seguir utilizando su aplicación en otras regiones. El problema durante la replicación de la aplicación es que también requiere que se repliquen los recursos. No solo necesita registrar todos los recursos que requiere su aplicación, sino que además debe aprovisionar y configurar esos recursos en cada región.

Vuelva a utilizar la plantilla de CloudFormation para crear sus recursos de una manera consistente y repetible. Para reutilizar sus plantillas, tiene que describir los recursos una vez y luego aprovisionar los mismos recursos una y otra vez en varias regiones.

Control y seguimiento sencillos de los cambios en la infraestructura

En algunos casos, es posible que tenga recursos subyacentes que desee actualizar de forma incremental. Por ejemplo, puede pasar a un tipo de instancia de mayor rendimiento en su configuración de lanzamiento de Auto Scaling, por lo que puede reducir el número máximo de instancias en su grupo de Auto Scaling. Si se producen problemas después de completar la actualización, es posible que necesite restaurar la configuración original de su infraestructura. Para hacerlo manualmente, no solo tiene que recordar qué recursos se han cambiado, sino también necesita conocer la configuración original.

Cuando aprovisiona su infraestructura con CloudFormation, la plantilla de CloudFormation describe exactamente qué recursos se aprovisionan y su configuración. Dado que estas plantillas son archivos de texto, solo tiene que realizar un seguimiento de las diferencias en las plantillas para realizar un seguimiento de los cambios en la infraestructura, de forma similar al modo en que los desarrolladores controlan las revisiones del código fuente. Por ejemplo, puede utilizar un sistema de control de versiones con las plantillas para saber exactamente qué cambios se han realizado, quién los ha realizado y cuándo. Si en cualquier momento necesita revertir los cambios de su infraestructura, puede utilizar una versión anterior de la plantilla.

Introducción a CloudFormation

CloudFormation está disponible a través de la [consola](#) de CloudFormation, la [API](#), [AWS CLI](#), [AWS SDK](#) y a través de varias integraciones.

Para obtener una introducción a CloudFormation, consulte [Funcionamiento de CloudFormation](#).

Para empezar a usar CloudFormation, consulte [Creación de su primera pila](#).

Información relacionada

Puede obtener más información sobre CloudFormation en esta guía del usuario, así como en los siguientes recursos:

- Para ver los detalles del producto y las preguntas frecuentes, consulte la [página de producto de AWS CloudFormation](#).

- Para obtener información sobre precios, consulte [Precios de AWS CloudFormation](#).

Introducción a CloudFormation

Puede empezar a utilizar CloudFormation a través de la Consola de administración de AWS mediante la creación de una pila a partir de una plantilla de ejemplo, que le ayudará a aprender los conceptos básicos de la creación de pilas. Una plantilla es un archivo de texto que define todos los recursos de una pila. Una pila es la implementación de una plantilla de CloudFormation. A partir de una sola plantilla, puede crear varias pilas. Cada pila contiene una colección de recursos de AWS, que se puede administrar como una única unidad.

CloudFormation es un servicio gratuito; no obstante, se le cobrará por los recursos de AWS que incluya en sus pilas a la tarifa actual para cada uno. Para obtener más información sobre los precios de AWS, consulte la página de detalles de cada producto en <http://aws.amazon.com>.

Temas

- [Funcionamiento de CloudFormation](#)
- [Registrarse para obtener Cuenta de AWS](#)
- [Creación de su primera pila](#)

Funcionamiento de CloudFormation

Este tema describe el funcionamiento de CloudFormation y le presenta los conceptos clave que necesitará conocer a medida que lo utilice.

Temas

- [Conceptos clave](#)
- [Funcionamiento de CloudFormation](#)
- [Formas de empezar con CloudFormation](#)

Conceptos clave

Cuando utiliza CloudFormation, trabaja con plantillas y pilas. Puede crear plantillas para describir los recursos de AWS y sus propiedades. Cuando se crea una pila, CloudFormation aprovisiona los recursos que se describen en su plantilla.

Temas

- [Plantillas](#)
- [Pilas](#)
- [Conjuntos de cambios](#)

Plantillas

Una plantilla de CloudFormation es un archivo de texto con formato JSON o YAML. Puede guardar estos archivos con cualquier extensión, como `.yaml`, `.json`, `.template` o `.txt`. CloudFormation usa estas plantillas como proyectos para crear sus recursos de AWS. Por ejemplo, en una plantilla, puede describir una instancia Amazon EC2, como el tipo de instancia, el ID de AMI, mapeos de dispositivos de bloques y el nombre de su par de claves de Amazon EC2. Cuando se crea una pila, debe especificarse una plantilla que CloudFormation utiliza para crear lo que se describe en la plantilla.

Por ejemplo, si creó una pila con la siguiente plantilla, CloudFormation aprovisiona una instancia con un ID de AMI `ami-0ff8a91507f77f867`, un tipo de instancia `t2.micro`, el nombre del par de claves `testkey` y un volumen de 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
          }
        }
      ]
    }
  }
}

```

También puede especificar varios recursos en una única plantilla y configurar dichos recursos para que funcionen conjuntamente. Por ejemplo, puede modificar la plantilla anterior para que incluya una dirección IP elástica (EIP) y asociarla a la instancia de Amazon EC2, tal y como se muestra en el ejemplo siguiente:

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"
        }
      }
    }
  }
}

```

```
}  
}
```

Las plantillas anteriores se centraron en una única instancia Amazon EC2; sin embargo, las plantillas de CloudFormation tienen capacidades adicionales que puede utilizar para crear conjuntos complejos de recursos y reutilizar esas plantillas en múltiples contextos. Por ejemplo, puede agregar parámetros de entrada cuyos valores se especifican al crear una pila de CloudFormation. En otras palabras, puede especificar un valor como el tipo de instancia al crear una pila en lugar de al crear la plantilla, lo que hace a la plantilla más fácil de reutilizar en diferentes situaciones.

Pilas

Al utilizar CloudFormation, los recursos relacionados se administran como una sola unidad, denominada pila. Puede crear, actualizar y eliminar una colección de recursos mediante la creación, actualización y eliminación de pilas. Todos los recursos de una pila se definen mediante la plantilla de CloudFormation de la pila. Supongamos que ha creado una plantilla que incluye un grupo de Auto Scaling, un balanceador de carga Elastic Load Balancing y una instancia de base de datos Amazon Relational Database Service (Amazon RDS). Para crear esos recursos, debe crear una pila enviando la plantilla que ha creado y CloudFormation aprovisiona todos esos recursos para usted.

Conjuntos de cambios

Si necesita realizar cambios en los recursos que están ejecutándose en una pila, actualice la pila. Antes de realizar cambios en los recursos, puede generar un conjunto de cambios, que es un resumen de los cambios propuestos. Los conjuntos de cambios le permiten ver cómo afectan los cambios a sus recursos en ejecución, en especial para los recursos críticos, antes de implementarlos.

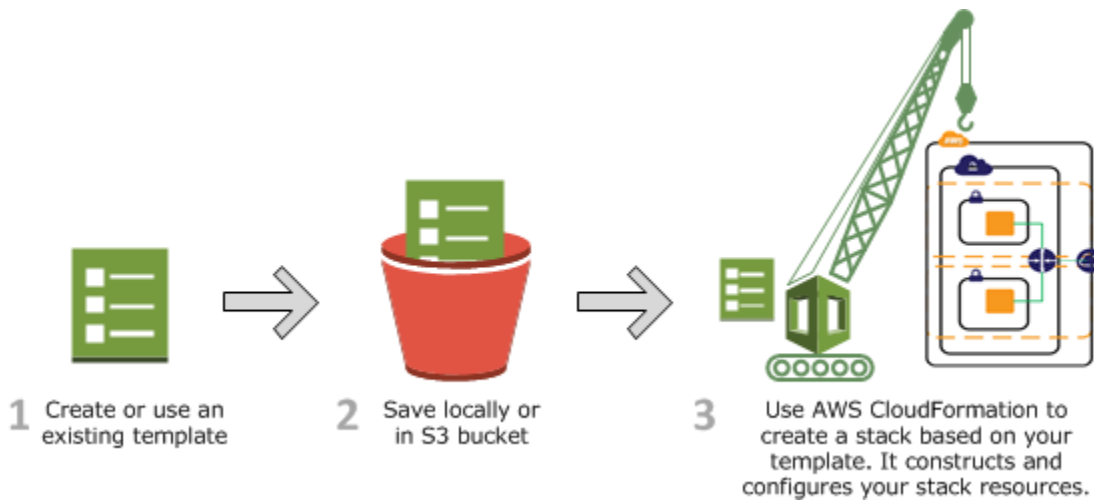
Por ejemplo, si cambia el nombre de una instancia de base de datos Amazon RDS, CloudFormation creará una nueva base de datos y eliminará la antigua. Se perderán los datos de la antigua base de datos a menos que ya haya hecho una copia de seguridad. Si genera un cambio conjunto, verá que el cambio provocará la sustitución de la base de datos y usted podrá planificar en consecuencia antes de actualizar la pila.

Funcionamiento de CloudFormation

Cuando usa CloudFormation para crear una pila, CloudFormation hace llamadas de servicio subyacentes a AWS para aprovisionar y configurar los recursos descritos en la plantilla. Necesita permiso para crear estos recursos. Por ejemplo, para crear instancias de EC2 mediante

CloudFormation, necesita permisos para crear instancias. Estos permisos se administran con [AWS Identity and Access Management](#) (IAM).

Su plantilla declara todas las llamadas que CloudFormation hace. Por ejemplo, suponga que tiene una plantilla que describe una instancia de EC2 con un tipo de instancia `t2.micro`. Cuando utiliza la plantilla para crear una pila, CloudFormation llama a la API de creación de instancias Amazon EC2 y especifica el tipo de instancia como `t2.micro`. En el siguiente diagrama se resume el flujo de trabajo de CloudFormation para la creación de pilas.



Para crear una pila

1. Use un editor de texto para crear una plantilla de CloudFormation en formato JSON o YAML. La CloudFormation plantilla describe los recursos que desea y sus ajustes. Se utiliza [Infrastructure Composer](#) para visualizar y validar la plantilla. Esto le ayuda a asegurarse de que la plantilla está estructurada correctamente y no contiene errores de sintaxis. Para obtener más información, consulte [Trabajo con plantillas de CloudFormation](#).
2. Guarde la plantilla de manera local o en un bucket de Amazon S3.
3. Cree una pila de CloudFormation especificando la ubicación del archivo de la plantilla; por ejemplo, una ruta en el equipo local o una URL de Amazon S3. Si la plantilla contiene parámetros, puede especificar los valores de entrada al crear la pila. Los parámetros le permiten pasar valores a la plantilla para poder personalizar sus recursos cada vez que crea una pila.

Note

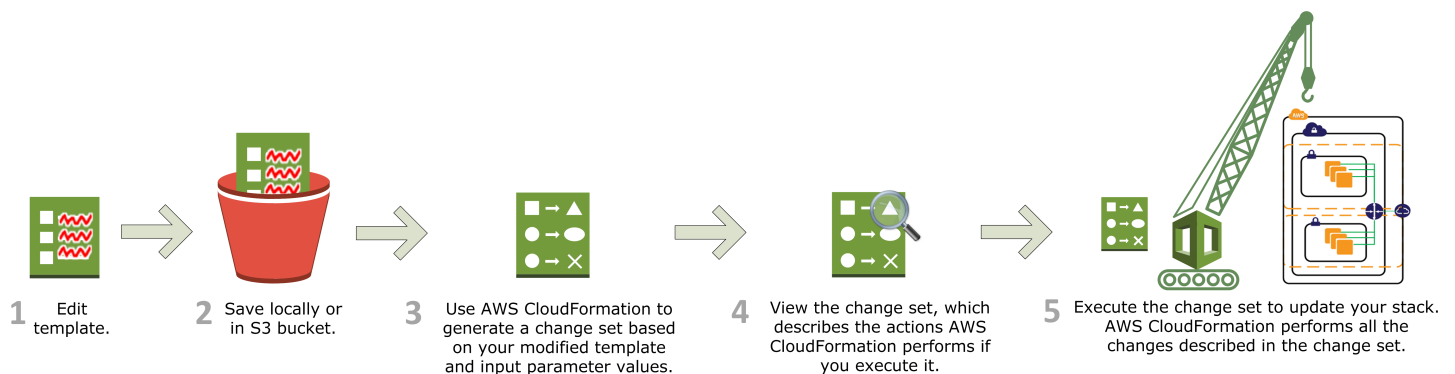
Si especifica un archivo de plantilla almacenado localmente, CloudFormation lo carga a un bucket de S3 en su cuenta de AWS. CloudFormation crea un bucket para cada región en la que carga un archivo de plantilla. Cualquiera con permisos de Amazon Simple

Storage Service (Amazon S3) en su cuenta de AWS puede acceder a los buckets. Si ya hubiera un bucket creado por CloudFormation, la plantilla se añade a dicho bucket. Puede utilizar su propio bucket y administrar sus permisos cargando manualmente las plantillas a Amazon S3. Entonces, siempre que cree o actualice una pila, especifique la URL de Amazon S3 de un archivo de plantilla.

Una vez que se han creado todos los recursos, CloudFormation informa que se ha creado su pila. A continuación, puede comenzar a utilizar los recursos de la pila. Si se produce un error durante la creación de la pila, CloudFormation revierte los cambios eliminando los recursos que creó.

Actualización de una pila con un conjunto de cambios

Cuando tenga que actualizar los recursos de su pila, puede modificar la plantilla de los recursos. No es necesario crear una nueva pila y eliminar la antigua. Para actualizar una pila, cree un conjunto de cambios enviando una versión modificada de la plantilla de pila original, diferentes valores de parámetros de entrada o ambos. CloudFormation compara la plantilla modificada con la original y genera un conjunto de cambios. El conjunto de cambios enumera las modificaciones propuestas. Tras revisar los cambios, puede iniciar el conjunto de cambios para actualizar la pila o puede crear un nuevo conjunto de cambios. En el siguiente diagrama se resume el flujo de trabajo para la actualización de una pila.



Actualización de una pila con un conjunto de cambios

1. Puede modificar una plantilla de pila de CloudFormation por medio de [Infrastructure Composer](#) o un editor de texto. Para obtener más información, consulte [Actualizar la plantilla de pila](#).

Al actualizar la plantilla, tenga en cuenta que las actualizaciones pueden provocar interrupciones. Dependiendo del recurso y de las propiedades que esté actualizando, una actualización podría interrumpir o incluso sustituir un recurso existente. Para obtener más

información, consulte [Comprensión de los comportamientos de actualización de los recursos de la pila](#).

2. Guarde la plantilla de CloudFormation localmente o en un bucket de S3.
3. Cree un conjunto de cambios especificando la pila que desea actualizar y la ubicación de la plantilla modificada, como por ejemplo una ruta en su equipo local o una dirección URL de Amazon S3. Para obtener más información sobre la creación de conjuntos de cambios, consulte [Actualización de pilas de CloudFormation con conjuntos de cambios](#).

 Note

Si especifica una plantilla almacenada en su equipo local, CloudFormation cargará automáticamente su plantilla a un bucket de S3 en su Cuenta de AWS.

4. Vea el conjunto de cambios para comprobar que CloudFormation realizará los cambios que espera. Por ejemplo, compruebe si CloudFormation reemplazará cualquier recurso crítico de la pila. Puede crear tantos conjuntos de cambios como necesite hasta que haya incluido los cambios que desea.

 Important

Los conjuntos de cambios no indican si la actualización de la pila se realizará correctamente. Por ejemplo, un conjunto de cambios no comprueba si se va a sobrepasar una [cuota](#) de cuenta, si actualizará un recurso que no admite actualizaciones o si no tiene suficientes [permisos](#) para modificar un recurso. Todo esto puede causar errores en la actualización de una pila.

5. Inicia el conjunto de cambios que desea aplicar a la pila. CloudFormation actualiza la pila mediante la actualización exclusiva de los recursos que ha modificado y señala que la pila se actualizó correctamente. Si se produce un error en la actualización de la pila, CloudFormation revierte los cambios para restaurar la pila al último estado de funcionamiento conocido.

Formas de empezar con CloudFormation

Para crear una pila de CloudFormation Hello world con la consola, consulte [Creación de su primera pila](#).

Para un aprendizaje guiado, pruebe el taller [Introducción a CloudFormation](#), que ofrece experiencia práctica en el desarrollo de plantillas.

Registrarse para obtener Cuenta de AWS

Al registrarse en AWS, su Cuenta de AWS se registra automáticamente en todos los servicios de AWS, incluido CloudFormation. Si ya dispone de una Cuenta de AWS, pase al siguiente tema. Si no dispone de una Cuenta de AWS, utilice el siguiente procedimiento para crear una.

Cómo crear una Cuenta de AWS

Si no dispone de una Cuenta de AWS, siga estos pasos para crear una.

Procedimiento para registrarse en Cuenta de AWS

1. Abra <https://portal.aws.amazon.com/billing/signup>.
2. Siga las instrucciones que se le indiquen.

Parte del procedimiento de registro consiste en recibir una llamada telefónica o mensaje de texto e indicar un código de verificación en el teclado del teléfono.

Al registrarse en una Cuenta de AWS, se crea un Usuario raíz de la cuenta de AWS. El usuario raíz tendrá acceso a todos los Servicios de AWS y recursos de esa cuenta. Como práctica recomendada de seguridad, asigne acceso administrativo a un usuario y utilice únicamente el usuario raíz para realizar [tareas que requieren acceso de usuario raíz](#).

AWS le enviará un correo electrónico de confirmación cuando complete el proceso de registro. Puede ver la actividad de la cuenta y administrarla en cualquier momento entrando en <https://aws.amazon.com/> y seleccionando Mi cuenta.

Creación de un usuario con acceso administrativo

Después de registrarse para obtener una Cuenta de AWS, proteja su Usuario raíz de la cuenta de AWS, habilite AWS IAM Identity Center y cree un usuario administrativo para no usar el usuario raíz en las tareas cotidianas.

Protección de Usuario raíz de la cuenta de AWS

1. Inicie sesión en [Consola de administración de AWS](#) como propietario de la cuenta; para ello, elija Usuario raíz e introduzca el correo electrónico de su Cuenta de AWS. En la siguiente página, escriba su contraseña.

Para obtener ayuda para iniciar sesión con el usuario raíz, consulte [Iniciar sesión como usuario raíz](#) en la Guía del usuario de AWS Sign-In.

2. Active la autenticación multifactor (MFA) para el usuario raíz.

Para obtener instrucciones, consulte [Habilitación de un dispositivo MFA virtual para su usuario raíz de la Cuenta de AWS \(consola\)](#) en la Guía del usuario de IAM.

Creación de un usuario con acceso administrativo

1. Activar IAM Identity Center.

Consulte las instrucciones en [Activar AWS IAM Identity Center](#) en la Guía del usuario de AWS IAM Identity Center.

2. En IAM Identity Center, conceda acceso administrativo a un usuario.

Para ver un tutorial sobre cómo usar Directorio de IAM Identity Center como origen de identidad, consulte [Configuración del acceso de los usuarios con el Directorio de IAM Identity Center predeterminado](#) en la Guía del usuario de AWS IAM Identity Center.

Inicio de sesión como usuario con acceso de administrador

- Para iniciar sesión con el usuario de IAM Identity Center, use la URL de inicio de sesión que se envió a la dirección de correo electrónico cuando creó el usuario de IAM Identity Center.

Para obtener ayuda para iniciar sesión con un usuario de IAM Identity Center, consulte [Inicio de sesión en el portal de acceso de AWS](#) en la Guía del usuario de AWS Sign-In.

Concesión de acceso a usuarios adicionales

1. En IAM Identity Center, cree un conjunto de permisos que siga la práctica recomendada de aplicar permisos de privilegios mínimos.

Para conocer las instrucciones, consulte [Create a permission set](#) en la Guía del usuario de AWS IAM Identity Center.

2. Asigne usuarios a un grupo y, a continuación, asigne el acceso de inicio de sesión único al grupo.

Para conocer las instrucciones, consulte [Add groups](#) en la Guía del usuario de AWS IAM Identity Center.

Note

Para obtener más información sobre cómo gestionar quién tiene acceso a qué, consulte [Control del acceso a CloudFormation con AWS Identity and Access Management](#).

Creación de su primera pila

En este tema se explica cómo crear su primera pila de CloudFormation con la Consola de administración de AWS. Si sigue este tutorial, obtendrá información sobre cómo aprovisionar recursos básicos de AWS, supervisar los eventos de la pila y generar resultados.

Para este ejemplo, la plantilla de CloudFormation está escrita en YAML. YAML es un formato legible por humanos que se usa ampliamente para definir la infraestructura como código. Al obtener más información sobre CloudFormation, es posible que también encuentre otras plantillas en formato JSON, pero para este tutorial, se elige YAML por su legibilidad.

Note

CloudFormation es gratuito, pero se le cobrarán los recursos de Amazon EC2 y Amazon S3 que cree. Sin embargo, si es la primera vez que utiliza AWS, puede aprovechar el [nivel gratuito](#) para minimizar o eliminar los costos durante este proceso de aprendizaje.

Temas

- [Requisitos previos](#)
- [Creación de una pila de CloudFormation con la consola](#)
- [Supervisión de la creación de una pila](#)

- [Prueba del servidor web](#)
- [Solución de problemas](#)
- [Limpieza](#)
- [Siguiendo pasos](#)

Requisitos previos

- Debe tener acceso a una Cuenta de AWS con un usuario o rol de IAM que tenga permisos para usar Amazon EC2, Amazon S3 y CloudFormation, o acceso de usuario administrativo.
- Debe tener una nube privada virtual (VPC) que tenga acceso a Internet. Para la plantilla de este tutorial se necesita una VPC predeterminada, que se incluye automáticamente con las Cuentas de AWS más recientes. Si no tiene una VPC predeterminada, o si se eliminó, consulte la sección de solución de problemas de este tema para ver soluciones alternativas.

Creación de una pila de CloudFormation con la consola

Creación de una pila de CloudFormation Hello world con la consola

1. Abra la [consola de CloudFormation](#).
2. Elija Crear pila.
3. En la página Crear pila, seleccione Crear a partir de Infrastructure Composer y, a continuación, Cree en Infrastructure Composer. De esta forma, accederá a Infrastructure Composer en el modo de consola de CloudFormation, donde puede cargar y validar la plantilla de ejemplo.
4. Para cargar y validar la plantilla de ejemplo, haga lo siguiente:
 - a. Elija Plantilla. A continuación, copie y pegue la siguiente plantilla de CloudFormation en el editor de plantillas:

```
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

Antes de avanzar al siguiente paso, dediquemos un momento a echar un vistazo a la plantilla y comprender algunos conceptos clave de CloudFormation.

- En la sección **Parameters** se declaran valores que se pueden transferir a la plantilla cuando crea la pila. Los recursos especificados más adelante en la plantilla hacen referencia a estos valores y utilizan los datos. Los parámetros son un método efectivo para especificar información que no desea almacenar en la propia plantilla. También sirven para especificar información que podría ser única de la configuración o de la aplicación específica que está implementando.
- La plantilla define los siguientes parámetros:
 - **LatestAmiId**: recupera el ID de la AMI de Amazon Linux 2 más reciente del Almacén de parámetros de AWS Systems Manager.
 - **InstanceType**: permite seleccionar el tipo de instancia de EC2 (valor predeterminado: t2.micro, valores permitidos: t3.micro, t2.micro).
 - **MyIP**: especifica el rango de direcciones IP para el acceso HTTP (el valor predeterminado es 0.0.0.0/0, que permite acceder desde cualquier IP).
- La sección **Resources** contiene las definiciones de los recursos de AWS que desea crear con la plantilla. Las declaraciones de recursos son una forma eficaz de especificar todos estos ajustes de configuración a la vez. Cuando incluye declaraciones de recursos en una plantilla, puede crear y configurar todos los recursos declarados si utiliza la plantilla para crear una pila. También puede crear nuevas pilas a partir de la misma plantilla para lanzar configuraciones de recursos idénticas.
- Esta plantilla crea los siguientes recursos:
 - **WebServerSecurityGroup**: grupo de seguridad de EC2 que permite el tráfico HTTP de entrada en el puerto 80 desde el rango de IP especificado.
 - **WebServer**: instancia de EC2 con la siguiente configuración:

- Usa la AMI de Amazon Linux 2 más reciente.
 - Aplica el tipo de instancia seleccionado.
 - Agrega `WebServerSecurityGroup` a la propiedad `SecurityGroupIds`.
 - Incluye un script de datos de usuario para instalar Apache HTTP Server.
 - Se especifica un nombre lógico al principio de cada declaración de recursos y parámetros. Por ejemplo, `WebServerSecurityGroup` es el nombre lógico asignado al recurso del grupo de seguridad de EC2. A continuación, la función `Ref` se utiliza para hacer referencia a los recursos y parámetros mediante sus nombres lógicos en otras partes de la plantilla. Cuando un recurso hace referencia a otro, se crea una dependencia entre ellos.
 - En la sección **Outputs** se definen los valores personalizados que se devuelven después de crear la pila. Puede usar los valores de salida para devolver información de los recursos de la pila, como identificadores de recursos o URL.
 - La plantilla define una salida:
 - **WebsiteURL**: URL del servidor web implementado, que se crea con el nombre de DNS público de la instancia de EC2. La función `Join` ayuda a combinar el parámetro fijo `http://` con el parámetro variable `PublicDnsName` en una sola cadena, lo que facilita la obtención de la URL completa del servidor web.
- b. Seleccione `Validate` para asegurarse de que el código YAML sea válido antes de cargar la plantilla.
 - c. A continuación, seleccione `Create template` para crear la plantilla y agregarla a un bucket de S3.
 - d. Tome nota del nombre del bucket de S3 que aparecen en el cuadro de diálogo que se abre para poder eliminarlo posteriormente. A continuación, seleccione `Confirm and continue to CloudFormation`. De esta forma, accede a la consola de CloudFormation, donde ahora se especifica la ruta de S3 a su plantilla.
5. En la página `Crear pila`, seleccione `Siguiente`.
 6. En la página `Especificar los detalles de la pila`, escriba un nombre en el campo `Nombre de la pila`. El nombre de la pila no debe contener espacios. Para este ejemplo, use **MyTestStack**.
 7. En `Parámetros`, especifique los valores de los parámetros de la siguiente manera:
 - `LatestAmild`: se define de forma predeterminada en la AMI de Amazon Linux 2 más reciente.
 - `InstanceType`: elija `t2.micro` o `t3.micro` para el tipo de instancia de EC2.

 Note

Si es la primera vez que utiliza AWS, puede usar el nivel gratuito para lanzar y usar una instancia `t2.micro` de forma gratuita durante 12 meses (en regiones donde `t2.micro` no esté disponible, puede usar una instancia `t3.micro` del nivel gratuito).

- MyIP: especifique su dirección IP pública real con un sufijo `/32`. El sufijo `/32` se usa en la notación CIDR para especificar que se permite una sola dirección IP. Básicamente, esto significa permitir el tráfico solo hacia y desde esta dirección IP específica.
8. Seleccione Siguiente dos veces para ir a la página Revisar y crear. Para este tutorial, puede dejar los valores predeterminados en la página Configurar opciones de pila tal y como están.
 9. Revisión de la información de la pila. Cuando esté satisfecho con la configuración, haga clic en Submit (Enviar).

Supervisión de la creación de una pila

Al seleccionar Enviar, CloudFormation comienza a crear los recursos que se especifican en la plantilla. La nueva pila, **MyTestStack**, aparece en la lista en la parte superior de la consola de CloudFormation. El estado debería ser `CREATE_IN_PROGRESS`. Puede ver el estado detallado de una pila visualizando sus eventos.

Visualización de los eventos de la pila

1. En la consola de CloudFormation, seleccione la pila **MyTestStack** en la lista.
2. En el panel de detalles de la pila, elija la pestaña Events (Eventos).

La consola actualiza automáticamente la lista de eventos con los últimos eventos cada 60 segundos.

La pestaña Events (Eventos) muestra todos los pasos importantes de la creación de la pila ordenados por la hora de cada evento, con los últimos eventos en la parte superior.

El primer evento (en la parte inferior de la lista de eventos) es el comienzo del proceso de creación de la pila:

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

A continuación se encuentran los eventos que señalan el comienzo y la finalización de la creación de cada recurso. Por ejemplo, la creación de la instancia de EC2 produce las siguientes entradas:

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

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

El evento `CREATE_IN_PROGRESS` se registra cuando CloudFormation informa que ha comenzado a crear el recurso. El evento `CREATE_COMPLETE` se registra cuando se crea correctamente el recurso.

Una vez que CloudFormation haya creado correctamente la pila, verá el siguiente evento en la parte superior de la pestaña Events (Eventos):

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

Si CloudFormation no puede crear un recurso, informa de un evento de `CREATE_FAILED` y, de forma predeterminada, restaura la pila y elimina todos los recursos que se han creado. La columna Status Reason (Motivo del estado) muestra el problema que ha provocado el error.

Una vez creada la pila, puede ir a la pestaña Recursos para ver la instancia de EC2 y el grupo de seguridad que creó.

Prueba del servidor web

Una vez que la pila se haya creado correctamente, vaya a la pestaña Salidas de la consola de CloudFormation. Busque el campo WebsiteURL. Contendrá la URL pública de su instancia de EC2.

Abra un navegador y vaya a la URL que aparece en WebsiteURL. Debería ver un mensaje sencillo ("Hello World!") en el navegador.

De esta forma, se confirma que su instancia de EC2 ejecuta Apache HTTP Server y presenta una página web básica.

Solución de problemas

Si se produce una reversión durante la creación de la pila, es posible que se deba a la falta de una VPC. Así es como se resuelve este problema.

No hay una VPC predeterminada disponible

Para la plantilla de este tutorial se necesita una VPC predeterminada. Si la creación de la pila falla debido a errores de disponibilidad de VPC o subred, es posible que no tenga una VPC predeterminada en su cuenta. Dispone de las opciones siguientes:

- Crear una nueva VPC predeterminada: puede crear una nueva VPC predeterminada mediante la consola de Amazon VPC. Para obtener instrucciones, consulte [Crear una VPC predeterminada](#) en la Guía del usuario de Amazon VPC.
- Modificar la plantilla para especificar una subred: si tiene una VPC no predeterminada, puede modificar la plantilla para especificar explícitamente los ID de la VPC y la subred. Agregue el siguiente parámetro a la plantilla:

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

A continuación, actualice el recurso `WebServer` para incluir el ID de la subred:

```
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
```

Al crear la pila, tendrá que especificar una subred que tenga acceso a Internet para que el servidor web sea accesible.

Limpieza

Para asegurarse de que no se le cobra por ningún servicio no deseado, puede realizar una limpieza eliminando la pila y sus recursos. También puede eliminar el bucket de S3 que almacena la plantilla de la pila.

Eliminación de la pila y sus recursos

1. Abra la [consola de CloudFormation](#).
2. En la página Pilas, seleccione la opción situada junto al nombre de la pila que ha creado (**MyTestStack**) y, a continuación, seleccione Eliminar.
3. Cuando se le pida confirmación, seleccione Eliminar.
4. Supervise el progreso del proceso de eliminación de la pila en la pestaña Evento. El estado de **MyTestStack** cambia a DELETE_IN_PROGRESS. Cuando CloudFormation completa la eliminación de la pila, quita la pila de la lista.

Si ha terminado de trabajar con la plantilla de ejemplo y ya no necesita el bucket de Amazon S3, elimínelo. Para eliminar un bucket, primero debe vaciarlo. Al vaciar un bucket, se eliminan todos los objetos que incluye.

Vaciado y eliminación del bucket de Amazon S3

1. Abra la [consola de Amazon S3](#).
2. En el panel de navegación del lado izquierdo de la consola, elija Buckets (Buckets).
3. En la lista Buckets, seleccione la opción situada junto al nombre del bucket que ha creado para este tutorial y, a continuación, seleccione Vaciar.
4. En la página Vaciar bucket, confirme que desea vaciar el bucket escribiendo **permanently delete** en el campo de texto y, a continuación, elija Vaciar.
5. Supervise el progreso del proceso de vaciado del bucket en la página Vaciado del bucket: estado.
6. Para volver a la lista de buckets, seleccione Salir.
7. Seleccione la opción situada junto al nombre del bucket y, a continuación, seleccione Eliminar.
8. Cuando se le pida la confirmación, escriba el nombre del bucket y, a continuación, seleccione Eliminar bucket.

9. Supervise el progreso del proceso de eliminación del bucket en la lista Buckets. Cuando Amazon S3 completa la eliminación del bucket, lo elimina de la lista.

Siguientes pasos

¡Enhorabuena! Ha creado correctamente una pila, ha supervisado su creación y ha utilizado su salida.

Para seguir aprendiendo:

- Obtenga más información sobre las plantillas para que pueda crear la suya propia. Para obtener más información, consulte [Trabajo con plantillas de CloudFormation](#).
- Pruebe el taller [Getting Started with CloudFormation](#) para adquirir práctica con la creación de plantillas.
- Para ver una versión abreviada de [Getting Started with CloudFormation](#), consulte [Implementación de aplicaciones en Amazon EC2](#). En este tema se describe el mismo escenario de uso de un script auxiliar de CloudFormation, `cfn-init`, para arrancar una instancia de Amazon EC2.

Prácticas recomendadas de CloudFormation

Las prácticas recomendadas son recomendaciones que pueden ayudarle a utilizar CloudFormation de forma más eficaz y a adoptar prácticas seguras a lo largo de todo su flujo de trabajo. Aprenda a planificar y organizar pilas, crear plantillas que describen los recursos y las aplicaciones de software que se ejecutan en ellas, y a administrar sus pilas y sus recursos. Las siguientes prácticas recomendadas se basan en experiencias reales de clientes de CloudFormation actuales.

Planificación y organización

- [Reduzca el bucle de retroalimentación para mejorar la velocidad de entrega](#)
- [Organice sus pilas por ciclo de vida y titularidad](#)
- [Utilice referencias entre pilas para devolver el valor de una salida exportada por otra pila](#)
- [Utilice StackSets de CloudFormation para implementaciones en varias cuentas y regiones](#)
- [Reutilice plantillas para replicar pilas en varios entornos](#)
- [Verifique las cuotas de todos los tipos de recursos](#)
- [Uso de módulos para reutilizar configuraciones de recursos](#)
- [Adopción de prácticas de infraestructura como código](#)

Creación de plantillas

- [No integre credenciales en sus plantillas](#)
- [Uso de tipos de parámetros específicos de AWS](#)
- [Uso de limitaciones de los parámetros](#)
- [Utilice pseudoparámetros para promover la portabilidad](#)
- [Use `AWS::CloudFormation::Init` para implementar aplicaciones de software en las instancias de Amazon EC2](#)
- [Utilice los scripts auxiliares más recientes](#)
- [Validar plantillas antes de usarlas](#)
- [Uso de YAML o JSON para la creación de plantillas](#)
- [Implemente una estrategia de etiquetado integral](#)
- [Aproveche las macros de plantillas para realizar transformaciones avanzadas](#)

Gestión de pilas

- [Administración de todos los recursos de pila a través de CloudFormation](#)

- [Cree conjuntos de cambios antes de actualizar las pilas](#)
- [Utilice políticas de pila para proteger los recursos.](#)
- [Uso de AWS CloudTrail para registrar llamadas a CloudFormation](#)
- [Utilice revisiones de código y controles de revisión para administrar las plantillas](#)
- [Actualice sus instancias de Amazon EC2 regularmente](#)
- [Utilice la detección de desviaciones regularmente.](#)
- [Configure activadores de reversión para la recuperación automática](#)
- [Implemente estrategias efectivas de refactorización de pilas](#)
- [Utilice CloudFormation Hooks para la administración del ciclo de vida](#)

Herramientas de creación

- [Utilice el generador IaC para crear plantillas a partir de recursos existentes.](#)
- [Uso de AWS Infrastructure Composer para el diseño visual de plantillas](#)
- [Considere utilizar AWS Cloud Development Kit \(AWS CDK\) para infraestructuras complejas](#)

Seguridad y conformidad

- [Utilice IAM para controlar el acceso](#)
- [Aplicación del principio de privilegios mínimos](#)
- [Proteja los parámetros confidenciales](#)
- [Implementación de la política como código con AWS CloudFormation Guard](#)

Reduzca el bucle de retroalimentación para mejorar la velocidad de entrega

Adopte prácticas y herramientas que le ayuden a acortar el bucle de retroalimentación de la infraestructura que describe con las plantillas de CloudFormation. Esto incluye realizar análisis y pruebas iniciales de las plantillas en su estación de trabajo. Cuando lo haga, tendrá la oportunidad de descubrir posibles problemas de sintaxis y configuración incluso antes de enviar sus contribuciones a un repositorio de código fuente. El descubrimiento temprano de estos problemas ayuda a evitar que lleguen a los entornos del ciclo de vida formal, como el desarrollo, el control de calidad y la producción. Este enfoque de pruebas tempranas y respuestas rápidas a errores le ofrece las ventajas de reducir el tiempo de espera para volver a trabajar, reducir las posibles áreas de impacto y aumentar su nivel de confianza a la hora de realizar operaciones de aprovisionamiento satisfactorias.

Las opciones de herramientas que lo ayudarán a desarrollar prácticas para responder rápido a los errores incluyen las herramientas de línea de comandos [CloudFormation Linter](#) (`cfn-lint`) y [TaskCat](#). La herramienta `cfn-lint` le permite validar sus plantillas de CloudFormation con la [especificación de recursos de CloudFormation](#). Esto incluye comprobar los valores válidos de las propiedades de los recursos, así como las prácticas recomendadas. Los complementos para `cfn-lint` están [disponibles para varios editores de código](#); esto le permite visualizar los problemas en su editor y obtener comentarios directos de Linter. También puede optar por integrar `cfn-lint` en la configuración de su repositorio de código fuente, de modo que pueda realizar la validación de la plantilla cuando confirme sus contribuciones. Para obtener más información, consulte [Validación previa a la confirmación de plantillas de CloudFormation de Git con cfn-lint](#). Una vez que haya realizado el lint inicial y solucionado cualquier problema que pudiera haber encontrado `cfn-lint`, puede usar TaskCat para probar sus plantillas al crear pilas mediante programación en las Regiones de AWS que elija. TaskCat también genera un informe con calificaciones de aprobación/reprobación para cada región que elija.

Para ver un tutorial práctico y paso a paso sobre cómo utilizar ambas herramientas para acortar el bucle de retroalimentación, siga el [Laboratorio de lint y pruebas](#) del [Taller de CloudFormation](#).

Organice sus pilas por ciclo de vida y titularidad

Utilice el ciclo de vida y la titularidad de sus recursos de AWS para ayudarlo a decidir qué recursos deberían ir en cada pila. Inicialmente, se pueden incluir todos los recursos en una sola pila, pero a medida que la pila crece en escala y su ámbito se amplía, administrar una sola pila puede ser tedioso y consumir mucho tiempo. Al agrupar los recursos con titularidad y ciclos de vida comunes, los propietarios pueden realizar cambios a su conjunto de recursos usando su propio proceso y programación sin que eso afecte a otros recursos.

Por ejemplo, imagine un equipo de desarrolladores e ingenieros que posee un sitio web alojado en instancias de Amazon EC2 Auto Scaling detrás de un equilibrador de carga. Dado que el sitio web tiene su propio ciclo de vida y lo mantiene su propio equipo, puede crear una pila para el sitio web y sus recursos. Ahora imagine que el sitio web también usa bases de datos de back-end, donde las bases de datos se encuentran en una pila aparte, bajo la titularidad y el mantenimiento de los administradores de bases de datos. Siempre que el equipo del sitio web o de la base de datos tengan que actualizar sus recursos, pueden hacerlo sin afectar a sus pilas mutuamente. Si todos los recursos estuvieran en una sola pila, coordinar y comunicar actualizaciones puede ser difícil.

Si desea obtener instrucciones adicionales sobre cómo organizar pilas, puede utilizar dos marcos de trabajo comunes: una arquitectura de varias capas y arquitectura orientada a servicios (SOA, por sus siglas en inglés).

Una arquitectura multicapa organiza pilas en varias capas horizontales apiladas unas sobre otras, y cada capa depende de la capa directamente inferior. Puede tener una o más pilas en cada capa, pero dentro de cada una, las pilas deberían tener recursos de AWS con ciclos de vida y titularidad similares.

Con una arquitectura orientada a los servicios, puede organizar grandes problemas empresariales en partes controlables. Cada una de estas partes es un servicio con una finalidad claramente definida y que representa una unidad autónoma de funcionalidad. Puede asignar estos servicios a una pila, de modo que cada pila tiene su propio ciclo de vida y propietarios. Estos servicios (pilas) pueden conectarse para que puedan interactuar.

Utilice referencias entre pilas para devolver el valor de una salida exportada por otra pila

Al organizar recursos de AWS en función del ciclo de vida y la titularidad, es posible que desee crear una pila que utiliza los recursos que están en otra pila. Puede codificar valores de forma predeterminada o usar parámetros de entrada para transferir ID y nombres de recursos. Sin embargo, con estos métodos las plantillas pueden ser difíciles de reutilizar o puede aumentar la sobrecarga para ejecutar una pila. En su lugar, utilice referencias entre pilas para devolver el valor de una salida exportada por otra pila para que otras pilas puedan utilizarlas. Las pilas pueden utilizar los recursos exportados llamándolos con la función `Fn::ImportValue`.

Por ejemplo, podría tener una pila de red que incluya una VPC, un grupo de seguridad y una subred. Le interesan todas las aplicaciones web públicas para utilizar estos recursos. Al exportar los recursos, permite que los usen todas las pilas con aplicaciones web públicas. Para obtener más información, consulte [Obtención de salidas exportadas de una pila implementada de CloudFormation](#).

Utilice StackSets de CloudFormation para implementaciones en varias cuentas y regiones

StackSets de CloudFormation amplía la funcionalidad de las pilas al permitirle crear, actualizar o eliminar pilas de varias cuentas y regiones con una sola operación. Utilice los StackSets para

implementar componentes de infraestructura comunes, controles de cumplimiento o servicios compartidos en toda su organización.

Cuando utilice StackSets, implemente permisos administrados por el servicio con AWS Organizations para simplificar la administración de permisos. Este enfoque permite implementar los StackSets en las cuentas de la organización sin necesidad de configurar manualmente las funciones de IAM en cada cuenta.

Para obtener más información, consulte [Conceptos de StackSets](#).

Verifique las cuotas de todos los tipos de recursos

Antes de lanzar una pila, asegúrese de que puede crear todos los recursos que desee sin llegar a los límites de su cuenta de AWS. En caso de llegar a un límite, CloudFormation no creará la pila de forma satisfactoria hasta que aumente su cuota o elimine recursos adicionales. Cada servicio puede tener varios límites que se deben tener en cuenta antes de lanzar una pila. Por ejemplo, de forma predeterminada, únicamente puede lanzar 2000 pilas de CloudFormation por región en su Cuenta de AWS. Para obtener más información acerca de los límites predeterminados, consulte [AWS Service Quotas](#) en la Referencia general de AWS.

Reutilice plantillas para replicar pilas en varios entornos

Después de configurar pilas y recursos, puede volver a utilizar las plantillas para replicar su infraestructura en varios entornos. Por ejemplo, puede crear entornos para desarrollo, pruebas y producción y así probar los cambios antes de implementarlos en la producción. Para que las plantillas sean reutilizables, utilice las secciones de parámetros, mapeos y condiciones para poder personalizar las pilas al crearlas. Por ejemplo, en el caso de los entornos de desarrollo, puede especificar un tipo de instancia de menor costo que la de su entorno de producción, pero el resto de configuraciones y valores seguirá igual. Para obtener más información acerca de los parámetros, mappings y condiciones, consulte [Secciones de la plantilla de CloudFormation](#).

Uso de módulos para reutilizar configuraciones de recursos

A medida que crece su infraestructura, pueden surgir patrones comunes donde usted declara los mismos componentes en cada plantilla. Los módulos son una forma de empaquetar configuraciones de recursos para su inclusión en plantillas de pila, de forma transparente, administrable y repetible. Los módulos pueden encapsular configuraciones de servicios comunes y prácticas recomendadas

como componentes esenciales modulares y personalizables para que pueda incluirlos en las plantillas de pila.

Estos bloques de construcción pueden destinarse a un único recurso, como las prácticas recomendadas para definir una instancia de Amazon Elastic Compute Cloud (Amazon EC2), o pueden servir para varios recursos, con el fin de definir patrones comunes de arquitectura de aplicaciones. Estos bloques de creación pueden anidarse en otros módulos, por lo que puede apilar prácticas recomendadas en bloques de creación de nivel superior. Los módulos de CloudFormation están disponibles en el [registro de CloudFormation](#) para que pueda usarlos como un recurso nativo. Cuando utiliza un módulo de CloudFormation, la plantilla del módulo se expande en la plantilla consumidora, lo que le permite acceder a los recursos dentro del módulo mediante [Ref](#) o [Fn::GetAtt](#). Para obtener más información, consulte [Creación de configuraciones de recursos reutilizables que se puedan incluir en las plantillas con los módulos de CloudFormation](#).

Adopción de prácticas de infraestructura como código

Trate sus plantillas de CloudFormation como código al implementar las prácticas de infraestructura como código (IaC). Almacene sus plantillas en sistemas de control de versiones, implemente revisiones de código y utilice pruebas automatizadas para validar los cambios. Este enfoque garantiza la coherencia, mejora la colaboración y proporciona un registro de auditoría de los cambios en la infraestructura.

Considere la posibilidad de implementar canalizaciones de CI/CD para el código de infraestructura a fin de automatizar las pruebas y la implementación de las plantillas de CloudFormation. Herramientas como AWS CodePipeline, AWS CodeBuild, y AWS CodeDeploy se pueden utilizar para crear flujos de trabajo automatizados para las implementaciones de su infraestructura.

Para obtener más información sobre la implementación de las prácticas recomendadas de IaC, consulte [Uso de AWS CloudFormation como herramienta de IaC](#).

Para obtener más información sobre el uso de la entrega continua con CloudFormation, consulte [Entrega continua con CodePipeline](#).

No integre credenciales en sus plantillas

En lugar de incrustar información confidencial en sus plantillas de CloudFormation, le recomendamos que utilice referencias dinámicas en su plantilla de pila.

Las referencias dinámicas son una forma eficaz y coherente de hacer referencia a valores externos almacenados y administrados en otros servicios, como AWS Systems Manager Parameter Store o AWS Secrets Manager. Cuando se utiliza una referencia dinámica, CloudFormation recupera el valor de la referencia especificada cuando es necesario durante las operaciones de pila y de conjunto de cambios, y pasa el valor al recurso correspondiente. Sin embargo, CloudFormation no almacena nunca el valor de referencia real. Para obtener más información, consulte [Uso de referencias dinámicas para especificar valores de plantillas](#).

[AWS Secrets Manager](#) ayuda a cifrar, almacenar y recuperar de forma segura las credenciales de sus bases de datos y otros servicios. [AWS Systems Manager Parameter Store](#) proporciona un almacenamiento seguro y jerárquico para administrar los datos de configuración.

Para obtener más información sobre cómo definir parámetros de plantillas, consulte [Sintaxis de Parameters de la plantilla de CloudFormation](#).

Uso de tipos de parámetros específicos de AWS

Si la plantilla requiere entradas para unos valores específicos de AWS existentes, como los ID de Amazon Virtual Private Cloud existentes o el nombre de un par de claves de Amazon EC2, use los tipos de parámetros específicos de AWS. Por ejemplo, puede especificar un parámetro como tipo `AWS::EC2::KeyPair::KeyName`, que adopta un nombre de par de claves existente que está en su Cuenta de AWS y en la región en la que se crea la pila. CloudFormation puede validar rápidamente los tipos de parámetros específicos de AWS antes de crear la pila. Además, si usa la consola de CloudFormation, mostrará una lista desplegable de valores válidos, lo que elimina la necesidad de buscar o memorizar los ID de VPC o nombres de pares de claves correctos. Para obtener más información, consulte [Definición de recursos existentes en tiempo de ejecución con tipos de parámetros proporcionados por CloudFormation](#).

Uso de limitaciones de los parámetros

Con limitaciones, puede describir los valores de entrada permitidos para que CloudFormation capture los valores no válidos antes de crear una pila. Puede establecer limitaciones como una longitud mínima, longitud máxima y patrones permitidos. Por ejemplo, puede establecer limitaciones en un valor de nombre de usuario de la base de datos, por lo que debe tener una longitud mínima de ocho caracteres y contener únicamente caracteres alfanuméricos. Para obtener más información, consulte [Sintaxis de Parameters de la plantilla de CloudFormation](#).

Utilice pseudoparámetros para promover la portabilidad

Puede utilizar [pseudoparámetros](#) en las plantillas como argumentos para [funciones intrínsecas](#), como Ref y Fn::Sub. Los pseudoparámetros son parámetros predefinidos por CloudFormation. No los declara en su plantilla. El uso de pseudoparámetros en las funciones intrínsecas aumenta la portabilidad de las plantillas de pilas entre regiones y cuentas.

Por ejemplo, imagine que desea crear una plantilla en la que, para una propiedad de recurso determinada, necesita especificar el [nombre de recurso de Amazon](#) (ARN) de otro recurso existente. En este caso, el recurso existente es un recurso del [almacén de parámetros de AWS Systems Manager](#) con el siguiente ARN: `arn:aws:ssm:us-east-1:123456789012:parameter/MySampleParameter`. Deberá adaptar el [formato del ARN](#) a su partición, región e ID de cuenta de AWS de destino. En lugar de realizar una codificación rígida de estos valores, puede usar los pseudoparámetros `AWS::Partition`, `AWS::Region` y `AWS::AccountId` para que su plantilla sea más portátil. En este caso, el siguiente ejemplo muestra cómo concatenar elementos de un ARN con CloudFormation: `!Sub 'arn:${AWS::Partition}:ssm:${AWS::Region}:${AWS::AccountId}:parameter/MySampleParameter'`.

Para otro ejemplo, suponga que desea compartir recursos o configuraciones entre varias pilas. En este ejemplo, suponga que ha creado una [subred](#) para la VPC y, a continuación, ha exportado su ID para usarlo con otras pilas de la misma Cuenta de AWS y región. En otra pila, hace referencia al valor exportado del ID de subred al describir una instancia de Amazon EC2. Para ver un ejemplo detallado del uso del campo de salida `Export` y la función intrínseca `Fn::ImportValue`, consulte [Consulta de las salidas de recursos en otra pila de CloudFormation](#).

Las exportaciones de pilas deben ser únicas por cuenta y por región. Por lo tanto, en este caso, puede usar el pseudoparámetro `AWS::StackName` para crear un prefijo para la exportación. Dado que los nombres de las pilas también deben ser únicos por cuenta y por región, el uso de este pseudoparámetro como prefijo aumenta la posibilidad de tener un nombre de exportación único y, al mismo tiempo, promueve un enfoque reutilizable en todas las pilas desde las que se exportan los valores. Otra alternativa es utilizar un prefijo de su elección.

Use **AWS::CloudFormation::Init** para implementar aplicaciones de software en las instancias de Amazon EC2

Al lanzar las pilas, puede instalar y configurar las aplicaciones de software en instancias de Amazon EC2 mediante el script auxiliar `cfn-init` y el recurso `AWS::CloudFormation::Init`. Al utilizar

`AWS::CloudFormation::Init`, puede describir las configuraciones que desee en lugar de unos pasos de procedimientos de scripts. También puede actualizar configuraciones sin volver a crear instancias. Y si algo no funciona con la configuración, CloudFormation genera registros que puede utilizar para investigar problemas.

En la plantilla, especifique los estados de instalación y configuración en el recurso

`AWS::CloudFormation::Init`. Para obtener una explicación detallada que muestra cómo utilizar `cfn-init` y `AWS::CloudFormation::Init`, consulte [Implementación de aplicaciones en Amazon EC2](#).

Utilice los scripts auxiliares más recientes

Los scripts auxiliares de CloudFormation se actualizan periódicamente. Asegúrese de incluir el siguiente comando en la propiedad `UserData` de la plantilla antes de llamar a los scripts auxiliares para garantizar que las instancias lanzadas reciban los últimos scripts auxliar:

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

Para obtener más información acerca de cómo obtener los scripts auxiliares más recientes, consulte [Referencia de scripts auxiliares de CloudFormation](#) en la Guía de referencia de plantillas de CloudFormation.

Validar plantillas antes de usarlas

Antes de utilizar una plantilla para crear o actualizar una pila, puede usar CloudFormation para validarla. Validar una plantilla puede ayudarle a identificar errores de sintaxis y algunos de semántica, como las dependencias circulares, antes de que CloudFormation cree recursos. Si usa la consola de CloudFormation, esta valida automáticamente la plantilla después de especificar parámetros de entrada. Para la AWS CLI o la API de CloudFormation, use el comando [validate-template](#) de la CLI o la operación [ValidateTemplate](#) de la API.

Durante la validación, CloudFormation comprueba primero si la plantilla es un JSON válido. Si no lo es, CloudFormation comprueba si la plantilla es un YAML válido. Si ambas comprobaciones fallan, CloudFormation devuelve un error de validación de plantilla.

Validar plantillas para la conformidad de políticas de organización

También puede validar su plantilla para comprobar el cumplimiento de las directrices de la política de la organización. AWS CloudFormation Guard (`cfn-guard`) es una herramienta de interfaz

de línea de comandos (CLI) de código abierto que proporciona un lenguaje de políticas como código para definir reglas que pueden comprobar las configuraciones de recursos obligatorias y prohibidas. A continuación, le permite validar sus plantillas en función de esas reglas. Por ejemplo, los administradores pueden crear reglas para asegurarse de que los usuarios siempre creen buckets de Amazon S3 cifrados.

Puede utilizar `cfn-guard` localmente, mientras edita plantillas o automáticamente como parte de una canalización CI/CD para parar la implementación de recursos no conformes.

Además, `cfn-guard` incluye una característica, `rulegen`, que permite extraer reglas de plantillas de CloudFormation compatibles existentes.

Para obtener más información, consulte el repositorio [cfn-guard](#) de GitHub.

Uso de YAML o JSON para la creación de plantillas

CloudFormation admite los formatos YAML y JSON para las plantillas. Cada uno tiene sus ventajas, y la elección depende de sus necesidades específicas:

Utilice YAML cuando

- Prioriza la legibilidad y la capacidad de mantenimiento humanas
- Quiere incluir comentarios para documentar su plantilla
- Está trabajando en plantillas complejas con estructuras anidadas
- Quiere utilizar características específicas de YAML, como anclajes y alias, para reducir la repetición

Utilice JSON cuando:

- Necesita integrarse con herramientas o sistemas que prefieran JSON
- Está trabajando con la generación o manipulación de plantillas programáticas
- Necesita una validación de datos estricta

Por lo general, se recomienda usar YAML para la creación manual de plantillas debido a su legibilidad y compatibilidad con los comentarios. Resulta especialmente útil para plantillas complejas en las que la estructura basada en indentaciones ayuda a visualizar las jerarquías de recursos. JSON puede resultar ventajoso en los flujos de trabajo automatizados o cuando se trabaja con API

que requieren la entrada de JSON. También es beneficioso cuando se necesita garantizar el estricto cumplimiento de una estructura específica. Independientemente del formato que elija, céntrese en crear plantillas bien estructuradas, documentadas y fáciles de mantener. Si utiliza YAML, aproveche sus características, como los anclajes y los alias, para reducir la repetición y mejorar la capacidad de mantenimiento.

Implemente una estrategia de etiquetado integral

Implemente una estrategia de etiquetado coherente para todos los recursos creados por sus plantillas de CloudFormation. Las etiquetas ayudan a organizar los recursos, asignar los costos, controlar el acceso y automatizar. Considere la posibilidad de incluir etiquetas para el entorno, el propietario, el centro de costos, la aplicación y el propósito.

Utilice la propiedad `Tags` del recurso `AWS::CloudFormation::Stack` para aplicar etiquetas a todos los recursos compatibles de una pila. También puede utilizar la propiedad `TagSpecifications` disponible en muchos tipos de recursos para aplicar etiquetas durante la creación del recurso.

Para obtener más información sobre el etiquetado, consulte [Etiquetar recursos](#).

Aproveche las macros de plantillas para realizar transformaciones avanzadas

Los macros de CloudFormation le permiten realizar procesamiento personalizado en las plantillas, desde acciones simples como operaciones de búsqueda y reemplazo hasta transformaciones complejas que generan recursos adicionales. Utilice macros para ampliar las capacidades de las plantillas de CloudFormation e implementar patrones reutilizables en toda la organización.

El AWS Serverless Application Model es un ejemplo de macro que simplifica el desarrollo de aplicaciones sin servidor. Considere la posibilidad de crear macros personalizadas para los patrones y requisitos específicos de la organización.

Para obtener más información sobre el uso de macros en las plantillas, consulte [Descripción general de las macros de CloudFormation](#).

Administración de todos los recursos de pila a través de CloudFormation

Después de lanzar una pila, use la [consola](#) de CloudFormation, la [API](#) o la [CLI de AWS](#) para actualizar los recursos de la pila. No realice cambios en los recursos de la pila fuera de CloudFormation. Eso puede crear una discrepancia entre la pila de la plantilla y el estado actual de los recursos de la pila, lo que puede producir errores si actualiza o elimina la pila. Esto se conoce como desviación. Si se realiza un cambio en un recurso fuera de la plantilla de CloudFormation y se actualiza la pila, los cambios realizados directamente en el recurso se descartarán y la configuración del recurso volverá a la configuración en la plantilla.

Para obtener más información sobre derivación, consulte [¿Qué es una desviación?](#).

Para obtener más información sobre la actualización de pilas, consulte [Actualización de una pila de CloudFormation](#).

Cree conjuntos de cambios antes de actualizar las pilas

Los conjuntos de cambios permiten ver cómo afectarían los cambios propuestos a una pila a los recursos que ya estén en ejecución, antes de implementarlos. CloudFormation no hace ningún cambio en la pila hasta que se ejecuta el conjunto de cambios, lo que le permite decidir si desea continuar con los cambios propuestos o crear otro conjunto de cambios.

Utilice conjuntos de cambio para comprobar cómo podrían afectar los cambios a los recursos en ejecución, especialmente para recursos de vital importancia. Por ejemplo, si cambia el nombre de una instancia de base de datos de Amazon RDS, CloudFormation creará una nueva base de datos y eliminará la antigua; se perderán los datos de la base antigua salvo que ya tenga una copia de seguridad. Si genera un conjunto de cambios, verá que el cambio sustituirá a la base de datos. Esto puede ayudarle a planificar antes de actualizar la pila. Para obtener más información, consulte [Actualización de pilas de CloudFormation con conjuntos de cambios](#).

Utilice políticas de pila para proteger los recursos.

Las políticas de pila ayudan a proteger los recursos de pila vitales de actualizaciones involuntarias que podrían interrumpir o incluso sustituir los recursos. Una política de pilas es un documento de JSON que describe qué acciones de actualización pueden realizarse en unos recursos designados. Especifique una política de pilas cada vez que cree una pila que tenga recursos de vital importancia.

Durante una actualización de pilas, debe especificar de forma explícita los recursos protegidos que desea actualizar; de lo contrario, no se producen cambios en los recursos protegidos. Para obtener más información, consulte [Impida actualizaciones en los recursos de la pila](#).

Uso de AWS CloudTrail para registrar llamadas a CloudFormation

AWS CloudTrail realiza un seguimiento a cualquier persona que realice llamadas a la API de CloudFormation en su Cuenta de AWS. Las llamadas a la API se registran cuando se utiliza la API de CloudFormation, la consola de CloudFormation, una consola de backend o comandos de la AWS CLI de CloudFormation. Active el registro y especifique un bucket de Amazon S3 para almacenar los registros. De ese modo, si lo necesita, puede auditar quién ha hecho cada llamada a CloudFormation en su cuenta.

Para obtener más información, consulte [Registro de llamadas a la API de CloudFormation con AWS CloudTrail](#).

Utilice revisiones de código y controles de revisión para administrar las plantillas

Sus plantillas de pila describen la configuración de los recursos de AWS, como los valores de las propiedades. Para revisar los cambios y mantener una historia exacta de los recursos, utilice revisiones de código y controles de revisión. Estos métodos pueden ayudarle a realizar un seguimiento de los cambios entre diferentes versiones de las plantillas, que pueden ayudarle a realizar un seguimiento de los cambios en los recursos de la pila. Además, al mantener un historial, siempre puede devolver la pila a una versión concreta de la plantilla.

Actualice sus instancias de Amazon EC2 regularmente

En todas sus instancias de Windows de Amazon EC2 y las instancias de Linux de Amazon EC2 creadas con CloudFormation ejecute el comando `yum update` con regularidad para actualizar el paquete RPM. Esto garantiza la obtención de las últimas correcciones y actualizaciones de seguridad.

Utilice la detección de desviaciones regularmente.

Utilice con regularidad la característica de detección de desviaciones de CloudFormation para identificar los recursos que se han modificado fuera de la administración de CloudFormation.

Detectar y resolver las desviaciones ayuda a mantener la integridad de la infraestructura como enfoque de código y garantiza que las plantillas reflejen con precisión el estado de los recursos implementados.

Considere la posibilidad de implementar la detección automática de desviaciones como parte de sus procedimientos operativos. Puede utilizar las características AWS Lambda activadas por las reglas de Amazon EventBridge para comprobar periódicamente si hay desviaciones y notificar a su equipo cuando se detecten discrepancias.

Para obtener más información sobre las desviaciones, consulte [Detectar cambios de configuración no administrados en pilas y recursos con detección de derivación](#).

Configure activadores de reversión para la recuperación automática

Utilice activadores de reversión para especificar las alarmas de Amazon CloudWatch que CloudFormation debe supervisar durante las operaciones de creación y actualización de las pilas. Si alguna de las alarmas especificadas entra en el estado ALARM, CloudFormation revierte automáticamente toda la operación de la pila, lo que ayuda a garantizar que la infraestructura permanezca en un estado estable.

Configure activadores de reversión para métricas críticas, como las tasas de error de las aplicaciones, la utilización de los recursos del sistema o las métricas empresariales personalizadas que indiquen el estado de su aplicación e infraestructura.

Para obtener más información sobre los activadores de reversión, consulte [Reversión de su pila ante una brecha de alarma](#).

Implemente estrategias efectivas de refactorización de pilas

A medida que su infraestructura evolucione, es posible que necesite refactorizar sus pilas de CloudFormation para mejorar la capacidad de mantenimiento, reducir la complejidad o adaptarse a los requisitos cambiantes. La refactorización de las pilas implica reestructurar las plantillas y los recursos y, al mismo tiempo, preservar su comportamiento y funcionalidad externos. La refactorización de pilas es beneficiosa si se utiliza con CloudFormation de las siguientes maneras:

- División de pilas monolíticas: desglose de pilas grandes y complejas en pilas más pequeñas y manejables, organizadas por ciclo de vida o propiedad

- Consolidación de los recursos relacionados: combinación de recursos relacionados de varias pilas en una sola pila cohesiva para simplificar la administración
- Extracción de componentes reutilizables: traslado de los patrones comunes a módulos o pilas anidadas para promover la reutilización y la coherencia
- Mejora de la organización de los recursos: reestructuración de los recursos dentro de una pila para reflejar mejor sus relaciones y dependencias

Para obtener más información sobre la refactorización de las pilas de CloudFormation, consulte [Refactorización de pilas](#).

Utilice CloudFormation Hooks para la administración del ciclo de vida

CloudFormation Hooks proporciona código que inspecciona de forma proactiva la configuración de los recursos de AWS antes del aprovisionamiento y realiza comprobaciones de validación complejas. Los Hooks comprueban si sus recursos, pilas y conjuntos de cambios cumplen con las necesidades de seguridad, operativas y de optimización de costos de su organización. Proporcionan advertencias antes del aprovisionamiento de un recurso o fallan en la operación y la detienen por completo, en función de cómo se haya configurado. Las infracciones y las advertencias se registran en Amazon CloudWatch para proporcionar visibilidad de las implementaciones no conformes.

Para obtener más información sobre estas prácticas recomendadas para Hooks, consulte [los conceptos de Hooks de AWS CloudFormation](#).

Para obtener más información sobre lo que Hooks puede hacer por sus recursos de CloudFormation, consulte [¿Qué son los Hooks de AWS CloudFormation?](#)

Utilice el generador IaC para crear plantillas a partir de recursos existentes.

El generador de la IaC (infraestructura como código) de CloudFormation le ayuda a crear plantillas de CloudFormation a partir de sus recursos de AWS existentes. Esta capacidad es particularmente útil cuando necesita replicar la infraestructura existente, documentar los recursos creados manualmente o poner recursos que antes no se administraban bajo la administración de CloudFormation. El generador de la IaC es útil para crear sus plantillas de CloudFormation de las siguientes maneras:

- Creación acelerada de plantillas: genere plantillas a partir de los recursos existentes en lugar de escribirlas desde cero
- Infraestructura coherente: asegúrese de que los nuevos entornos coincidan con los existentes utilizando las plantillas generadas como punto de partida
- Migración a la infraestructura como código: incorpore gradualmente los recursos creados manualmente a la administración de CloudFormation
- Documentación: cree un registro de su infraestructura existente en forma de plantilla

Para obtener más información sobre el generador de la IaC, consulte [Generación de plantillas de los recursos existentes con el generador de la IaC](#).

Uso de AWS Infrastructure Composer para el diseño visual de plantillas

AWS Infrastructure Composer es una herramienta de diseño visual que le ayuda a crear, visualizar y modificar plantillas de CloudFormation mediante una interfaz de arrastrar y soltar. Puede ser particularmente beneficioso cuando utiliza CloudFormation de las siguientes maneras:

- Planificación de la arquitectura: diseñe y valide las arquitecturas de infraestructura antes de la implementación
- Modernización de plantillas: visualice las plantillas existentes para comprender su estructura e identificar oportunidades de mejora
- Capacitación e incorporación: ayude a los nuevos miembros del equipo a entender los conceptos de CloudFormation y las relaciones de servicio de AWS mediante el aprendizaje visual
- Comunicación con las partes interesadas: presente los diseños de infraestructura a las partes interesadas no técnicas mediante representaciones visuales claras
- Revisiones de conformidad: utilice diagramas visuales para facilitar las revisiones de seguridad y conformidad de los diseños de su infraestructura
- Revisiones de conformidad: utilice diagramas visuales para facilitar las revisiones de seguridad y conformidad de los diseños de su infraestructura

Para obtener más información acerca de Infrastructure Composer, consulte [¿Qué es AWS Infrastructure Composer?](#).

Considere utilizar AWS Cloud Development Kit (AWS CDK) para infraestructuras complejas

Para requisitos de infraestructura complejos, considere utilizar el CDK para definir sus recursos en la nube utilizando lenguajes de programación familiares como TypeScript, Python, Java y .NET. AWS CDK genera plantillas de CloudFormation a partir de su código, lo que le permite aprovechar todas las capacidades de CloudFormation mientras utiliza las abstracciones y construcciones de programación de su lenguaje preferido.

AWS CDK proporciona constructos de alto nivel que resumen las prácticas recomendadas y simplifican la definición de patrones de infraestructura comunes. Esto puede reducir significativamente la cantidad de código necesaria para definir su infraestructura y, al mismo tiempo, garantizar el cumplimiento de las prácticas recomendadas.

Para más información sobre CDK, consulte [AWS Cloud Development Kit \(AWS CDK\)](#).

Utilice IAM para controlar el acceso

IAM es un servicio de AWS que puede utilizar para administrar los usuarios y sus permisos en AWS. Puede utilizar IAM con CloudFormation para especificar qué acciones de CloudFormation pueden realizar los usuarios, como ver plantillas de pila, crear pilas o eliminar pilas. Además, cualquiera que administre pilas de CloudFormation necesitará permisos para acceder a los recursos de dichas pilas. Por ejemplo, si los usuarios desean usar CloudFormation para lanzar, actualizar o terminar instancias Amazon EC2, deben tener permiso para llamar a las acciones de Amazon EC2 pertinentes.

En la mayoría de los casos, los usuarios requieren acceso completo para administrar todos los recursos de una plantilla. CloudFormation realiza llamadas para crear, modificar y eliminar esos recursos en su nombre. Para separar los permisos entre el usuario y el servicio de CloudFormation, utilice un rol de servicio. CloudFormation utiliza la política del rol de servicio para realizar llamadas en lugar de la política del usuario. Para obtener más información, consulte [Rol de servicio CloudFormation](#).

Aplicación del principio de privilegios mínimos

Al configurar los roles de IAM para los roles de servicio de CloudFormation o para los recursos creados por las plantillas, aplique siempre el principio de privilegios mínimos. Otorgue únicamente los

permisos necesarios para la funcionalidad prevista y evite utilizar permisos comodín siempre que sea posible.

Utilice IAM Access Analyzer para revisar los permisos concedidos a sus roles de servicio de CloudFormation e identificar los permisos no utilizados que se pueden eliminar. Revise y actualice regularmente las políticas de IAM para garantizar que permanezcan alineadas con sus requisitos de seguridad.

Proteja los parámetros confidenciales

Para información confidencial, como contraseñas, claves de API y otros secretos, utilice Parameter Store AWS Systems Manager o AWS Secrets Manager en lugar de incrustarlos directamente en sus plantillas. Utilice referencias dinámicas en sus plantillas para recuperar estos valores de forma segura durante las operaciones de apilado.

Cuando utilice parámetros en sus plantillas, establezca la propiedad NoEcho en `true` para los parámetros confidenciales para evitar que sus valores se muestren en la consola, en las respuestas de la API o en la salida de la CLI. Tenga en cuenta que NoEcho no impide que se registre el valor si se transfiere a otros servicios o recursos que puedan registrarlo.

Para obtener más información sobre el uso de Parameter Store AWS Systems Manager con CloudFormation, consulte [Obtener un valor de texto sin formato de Parameter Store AWS Systems Manager](#).

Para obtener más información sobre el uso de la propiedad NoEcho, consulte la [Sintaxis de los parámetros de la plantilla de CloudFormation](#).

Para obtener más información sobre el uso de AWS Secrets Manager con CloudFormation, consulte [Crear secretos AWS Secrets Manager en AWS CloudFormation](#).

Implementación de la política como código con AWS CloudFormation Guard

AWS CloudFormation Guard (`cfn-guard`) es una herramienta de código abierto de política como código que le permite definir y aplicar reglas para sus plantillas de CloudFormation. Utilice `cfn-guard` para asegurarse de que sus plantillas cumplan con las políticas organizativas, las prácticas recomendadas de seguridad y los requisitos de gobernanza.

Integre `cfn-guard` en sus procesos de CI/CD para validar automáticamente las plantillas según las normas de sus políticas antes de su implementación. Esto ayuda a evitar que se implementen recursos que no cumplan con las normas en su entorno y proporciona a los desarrolladores información temprana sobre las infracciones de las políticas.

Para obtener más información acerca de Guard, consulte [¿Qué es AWS CloudFormation Guard?](#)

Trabajo con plantillas de CloudFormation

Una plantilla de AWS CloudFormation define los recursos de AWS que desea crear, actualizar o eliminar como parte de una pila. Tiene varias secciones, pero la única necesaria es la sección de [Resources](#), que debe declarar al menos un recurso.

Puede crear plantillas mediante uno de los siguientes métodos:

- AWS Infrastructure Composer: una interfaz visual para diseñar plantillas.
- Editor de texto: escriba plantillas directamente en la sintaxis JSON o YAML.
- Generador de IaC: genere plantillas a partir de los recursos aprovisionados en su cuenta que CloudFormation no administra actualmente. El generador de IaC funciona con una amplia gama de recursos compatibles con la API de Cloud Control en su región.

En esta sección se proporciona una guía completa sobre cómo utilizar las distintas secciones de una plantilla de CloudFormation y cómo empezar a crear plantillas de pila. Abarca los siguientes temas:

Temas

- [Dónde se almacenan las plantillas](#)
- [Validación de plantillas](#)
- [Introducción a las plantillas](#)
- [Plantillas de ejemplo](#)
- [Formato de plantilla de CloudFormation](#)
- [Secciones de la plantilla de CloudFormation](#)
- [Creación de plantillas visualmente con Infrastructure Composer](#)
- [Servidor de lenguaje AWS CloudFormation](#)
- [Generar plantillas de los recursos existentes con el generador IaC](#)
- [Obtención de valores almacenados en otros servicios con referencias dinámicas](#)
- [Obtención de valores de AWS mediante pseudoparámetros](#)
- [Obtención de salidas exportadas de una pila implementada de CloudFormation](#)
- [Definición de recursos existentes en tiempo de ejecución con tipos de parámetros proporcionados por CloudFormation](#)
- [Tutoriales de CloudFormation](#)

- [Fragmentos de plantillas de CloudFormation](#)
- [Implementación de pilas basadas en Windows usando CloudFormation](#)
- [Amplíe las capacidades de su plantilla con los tipos de recursos proporcionados por CloudFormation](#)
- [Creación de configuraciones de recursos reutilizables que se puedan incluir en las plantillas con los módulos de CloudFormation](#)

Dónde se almacenan las plantillas

Bucket de Amazon S3

Puede almacenar plantillas de CloudFormation en un bucket de Amazon S3. Al crear o actualizar una pila, puede especificar la URL de S3 de la plantilla en lugar de cargarla directamente.

Si carga las plantillas directamente a través de Consola de administración de AWS o AWS CLI, automáticamente se creará un bucket de S3. Para obtener más información, consulte [Creación de una pila en la consola de CloudFormation](#).

Repositorio de Git

Con la [sincronización de Git](#), puede almacenar plantillas en un repositorio de Git. Al crear o actualizar una pila, puede especificar la ubicación y la rama del repositorio de Git que contiene la plantilla en lugar de subirla directamente o referenciar a una URL de S3. CloudFormation supervisa automáticamente el repositorio y la rama especificados para detectar cambios en las plantillas. Para obtener más información, consulte [Creación de una pila a partir del código fuente del repositorio con sincronización de Git](#).

Validación de plantillas

Validación de la sintaxis

Puede comprobar la sintaxis JSON o YAML de la plantilla con el comando [validate-template](#) de la CLI o al especificar la plantilla en la consola. La consola realiza la validación automáticamente. Para obtener más información, consulte [Creación de una pila en la consola de CloudFormation](#).

Sin embargo, estos métodos solo verifican la sintaxis de la plantilla y no validan los valores de propiedad que especificó para un recurso.

Herramientas de validación adicionales

Para validaciones más complejas y comprobaciones de prácticas recomendadas, puede usar herramientas adicionales como:

- [CloudFormation Linter \(cfn-lint\)](#): valida las plantillas con los [esquemas del proveedor de recursos de CloudFormation](#). Incluye comprobar los valores válidos de las propiedades de los recursos y las prácticas recomendadas.
- [CloudFormation Rain \(rain fmt\)](#): formatea sus plantillas de CloudFormation según un estándar coherente o cambia el formato de una plantilla de JSON a YAML (o de YAML a JSON). Conserva los comentarios cuando se usa YAML y, siempre que sea posible, cambia el uso de funciones intrínsecas a una sintaxis corta.

Introducción a las plantillas

Para empezar a crear una plantilla de CloudFormation, siga estos pasos:

1. Elegir los recursos: identifique los recursos de AWS que desea incluir en su pila, como las instancias de EC2, las VPC, los grupos de seguridad, etc.
2. Redactar la plantilla: escriba la plantilla en formato JSON o YAML y defina los recursos y sus propiedades.
3. Guardar la plantilla: guarde la plantilla localmente con una extensión de archivo como: `.json`, `.yaml` o `.txt`.
4. Validar la plantilla: valide la plantilla con los métodos descritos en la sección [Validación de plantillas](#).
5. Crear una pila: cree una pila con la plantilla validada.

Planee usar la plantilla de referencia de CloudFormation

A medida que vaya redactando las plantillas, encontrará documentación sobre la sintaxis detallada de los distintos tipos de recursos en la [referencia de tipos de recursos y propiedades de AWS](#).

A menudo, las plantillas de pila requieren funciones intrínsecas para asignar valores de propiedad que no están disponibles hasta el tiempo de ejecución y atributos especiales para controlar el comportamiento de los recursos. A medida que redacte la plantilla, consulte los siguientes recursos para obtener orientación:

- [Referencia de funciones intrínsecas](#): algunas de las funciones intrínsecas más utilizadas incluyen a las siguientes:
 - Ref: recupera el valor del parámetro o el ID físico de un recurso.
 - Sub: sustituye los marcadores de posición de las cadenas por valores reales.
 - GetAtt: devuelve el valor de un atributo de un recurso de la plantilla.
 - Join: une un conjunto de valores en una sola cadena.
- [Referencia de atributos de recursos](#): algunos de los atributos especiales más utilizados incluyen a las siguientes:
 - DependsOn: use este atributo para especificar que un recurso debe crearse después de otro.
 - DeletionPolicy: use este atributo para especificar cómo debe manejar CloudFormation la eliminación de un recurso.

Plantillas de ejemplo

CloudFormation proporciona plantillas de pila de código abierto que puede utilizar para empezar. Para obtener más información, consulte [Plantillas de CloudFormation de muestra](#) en el sitio web de GitHub.

Tenga en cuenta que no se pretende que estas plantillas estén listas para la producción. Debería tomarse el tiempo necesario para aprender cómo funcionan, adaptarlas a sus necesidades y asegurarse de que cumplan con los estándares de cumplimiento de su empresa.

Cada plantilla de este repositorio pasa las comprobaciones de [CloudFormation Linter](#) (cfn-lint) y también un conjunto básico de reglas AWS CloudFormation Guard basadas en las 20 principales de Center for Internet Security (CIS), con la excepción de algunas reglas en las que tenía sentido mantener la muestra centrada en un único caso de uso.

Formato de plantilla de CloudFormation

Puede crear plantillas de CloudFormation en formatos JSON o YAML. Ambos formatos tienen el mismo propósito, pero ofrecen claras ventajas en términos de legibilidad y complejidad.

- JSON: JSON es un formato de intercambio de datos ligero que las máquinas pueden analizar y generar fácilmente. Sin embargo, leer y escribir puede ser engorroso para los humanos, en especial en configuraciones complejas. En JSON, la plantilla se estructura mediante llaves {} y

corchetes `[]` anidados para definir los recursos, los parámetros y otros componentes. Su sintaxis requiere una declaración explícita de cada elemento, lo que puede hacer que la plantilla sea más detallada, pero garantiza el cumplimiento estricto de un formato estructurado.

- **YAML:** YAML está diseñado para ser más legible y menos detallado que JSON. Usa indentación en lugar de llaves y corchetes para indicar el anidamiento, lo que puede facilitar la visualización de la jerarquía de recursos y parámetros. A menudo se prefiere YAML por su claridad y facilidad de uso, en especial cuando se trata de plantillas más complejas. Sin embargo, el hecho de que YAML dependa de la indentación puede provocar errores si el espaciado no es uniforme, lo que requiere una atención cuidadosa para mantener la precisión.

Estructura de la plantilla

Las plantillas de CloudFormation se dividen en diversas secciones y cada una de ellas está diseñada para contener un tipo específico de información. Algunas secciones deben declararse en un orden específico y, en el caso de otras, el orden no importa. Sin embargo, al crear la plantilla, puede ser útil utilizar el orden lógico del siguiente ejemplo, ya que los valores de una sección podrían estar relacionados con los valores de una sección anterior.

Al crear plantillas, no use secciones principales duplicadas, por ejemplo, la sección `Resources`. Aunque CloudFormation puede aceptar la plantilla, esta puede tener un comportamiento indefinido al procesarla, lo que podría resultar en el aprovisionamiento incorrecto de recursos o en la devolución de errores inexplicables.

JSON

En el siguiente ejemplo se muestra la estructura de una plantilla con formato JSON con todas las secciones disponibles.

```
{
  "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

En el siguiente ejemplo se muestra la estructura de una plantilla en formato YAML con todas las secciones disponibles.

```
---  
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*

Comentarios

En las plantillas con formato JSON, no se admiten comentarios. JSON, por diseño, no incluye una sintaxis para los comentarios, lo que significa que no se pueden agregar comentarios directamente dentro de la estructura de JSON. Sin embargo, si necesita incluir notas explicativas o documentación, puede agregar metadatos. Para obtener más información, consulte [Metadato atributo](#).

En las plantillas con formato YAML, puede incluir comentarios en línea con el símbolo #.

El siguiente ejemplo muestra una plantilla YAML con comentarios en línea.

```
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
```

Especificaciones

CloudFormation es compatible con las siguientes especificaciones de JSON y YAML:

JSON

CloudFormation sigue el estándar JSON ECMA-404. Para obtener más información sobre el formato JSON, consulte <http://www.json.org>.

YAML

CloudFormation admite la especificación YAML versión 1.1 con algunas excepciones.

CloudFormation no admite las siguientes características:

- Las etiquetas `binary`, `omap`, `pairs`, `set` y `timestamp`
- Alias
- Fusiones de hash

Para obtener más información acerca de YAML, consulte <http://yaml.org/>.

Más información

Para cada recurso que especifique en su plantilla, debe definir sus propiedades y valores mediante las reglas de sintaxis específicas de JSON o YAML. Para obtener más información acerca de la sintaxis de la plantilla para cada formato, consulte [Secciones de la plantilla de CloudFormation](#).

Uso de expresiones regulares en plantillas de CloudFormation

Puede utilizar expresiones regulares (conocidas habitualmente como regexes) en una serie de lugares en sus plantillas de CloudFormation, como, por ejemplo, para la propiedad `AllowedPattern` al crear un [parámetro](#) de plantilla.

Todas las expresiones regulares en CloudFormation cumplen la sintaxis de expresiones regulares de Java. Para obtener una descripción completa de la sintaxis de expresiones regulares de Java y sus constructos, consulte java.util.regex.Pattern.

Si escribe la plantilla de CloudFormation en sintaxis JSON, debe aplicar escape cualquier carácter de barra invertida (\) en la expresión regular mediante la incorporación de una barra invertida adicional. Esto se debe a que JSON interpreta las barras invertidas como caracteres de escape, y necesita aplicarles escape para asegurarse de que se tratan como barras invertidas literales en la expresión regular.

Por ejemplo, si incluye una `\d` en su expresión regular para que coincida con un dígito, tendrá que escribirlo como `\\d` en su plantilla de JSON.

En el ejemplo siguiente, la propiedad `AllowedPattern` especifica una expresión regular que coincide con cuatro caracteres de dígitos consecutivos (`\d{4}`). Sin embargo, dado que la expresión regular se define en una plantilla JSON, es necesario aplicar escape al carácter de barra invertida con una barra invertida adicional (`\\d`).

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

Si escribe la plantilla de CloudFormation en la sintaxis YAML, debe escribir la expresión regular entre comillas simples ('). No se necesita aplicar una secuencia de escape adicional.

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

Note

Las expresiones regulares en CloudFormation solo se admiten con fines de validación en contextos específicos, como `AllowedPattern`. No se admiten como operaciones de coincidencia de patrones en las funciones intrínsecas de CloudFormation, como

`Fn::Equals`, que solo realiza una comparación exacta de cadenas, no una coincidencia de patrones.

Secciones de la plantilla de CloudFormation

Cada plantilla de CloudFormation consta de una o más secciones, cada una con un propósito específico.

La sección **Recursos** es obligatoria en todas las plantillas de CloudFormation y constituye el núcleo de la plantilla. Esta sección especifica los recursos de la pila y sus propiedades, como una instancia de Amazon EC2 o un bucket de Amazon S3. Cada recurso se define con un identificador lógico único, un tipo y detalles de configuración específicos.

La sección de **Parámetros**, si bien es opcional, desempeña un rol importante a la hora de hacer que las plantillas sean más flexibles. Permite a los usuarios pasar valores en tiempo de ejecución al crear o actualizar una pila. Se puede referenciar estos parámetros en las secciones **Resources** y **Outputs** y, de este modo, se puede personalizar sin alterar la propia plantilla. Por ejemplo, puede usar los parámetros para especificar los tipos de instancias o la configuración del entorno que varían de una implementación a otra.

La sección **Salidas**, también opcional, define los valores que se devuelven al ver las propiedades de una pila. Los resultados ofrecen información útil, como identificadores de recursos o URL, que se pueden utilizar con fines operativos o para integrarlos con otras pilas. Esta sección ayuda a los usuarios a recuperar y utilizar detalles importantes sobre los recursos creados por la plantilla.

Otras secciones opcionales incluyen las **Asignaciones**, que funcionan como tablas de búsqueda para administrar los valores condicionales. Con las asignaciones, se definen pares clave-valor y se utilizan con la función intrínseca `Fn::FindInMap` en las secciones **Resources** y **Outputs**. Esto resulta útil en situaciones en las que es necesario ajustar las configuraciones según condiciones tales como Región de AWS o el entorno.

Las secciones **Metadatos** y **Reglas**, aunque se utilizan con menos frecuencia, ofrecen funciones adicionales. **Metadata** puede incluir información adicional sobre la plantilla, mientras que **Rules** valida un parámetro o una combinación de parámetros durante la creación o las actualizaciones de la pila, lo que garantiza que se cumplan criterios específicos. La sección **Condiciones** mejora aún más la flexibilidad al controlar si determinados recursos se crean o si a las propiedades se les asigna un valor según condiciones como el tipo de entorno.

Por último, la sección Transformar se utiliza para aplicar macros durante el procesamiento de la plantilla. Para las aplicaciones sin servidor (también denominadas aplicaciones de Lambda), especifica la versión de [AWS Serverless Application Model \(AWS SAM\)](#) que hay que utilizar. Cuando especifique una transformación, puede utilizar sintaxis AWS SAM para declarar los recursos en la plantilla. El modelo define la sintaxis que puede utilizar y la forma en la que se procesa. También puede usar la transformación `AWS::Include` para incluir fragmentos de plantilla que se almacenan por separado de la plantilla principal de CloudFormation.

Los siguientes temas brindan más información y ejemplos de uso de cada sección.

Temas

- [Sintaxis de Resources de la plantilla de CloudFormation](#)
- [Sintaxis de Parameters de la plantilla de CloudFormation](#)
- [Sintaxis de Outputs de la plantilla de CloudFormation](#)
- [Sintaxis de Mappings de la plantilla de CloudFormation](#)
- [Sintaxis de Metadata de la plantilla de CloudFormation](#)
- [Sintaxis de Rules de la plantilla de CloudFormation](#)
- [Sintaxis de Conditions de la plantilla de CloudFormation](#)
- [Sección Transform de la plantilla de CloudFormation](#)
- [Sintaxis de la versión del formato de plantilla de CloudFormation](#)
- [Sintaxis de Description de la plantilla de CloudFormation](#)

Sintaxis de Resources de la plantilla de CloudFormation

La sección `Resources` es una sección de nivel superior obligatoria en una plantilla de CloudFormation. Declara los recursos de AWS que usted desea que CloudFormation aprovisiona y configure como parte de su pila.

Sintaxis

La sección `Resources` usa la siguiente sintaxis:

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
      ...

```

ID lógico (también denominado nombre lógico)

En una plantilla de CloudFormation, los recursos se identifican por sus nombres de recursos lógicos. Estos nombres deben ser alfanuméricos (A-Za-z0-9) y únicos dentro de la plantilla. Los nombres lógicos se usan para referenciar a los recursos de otras secciones de la plantilla.

Tipo de recurso

Cada recurso debe tener un atributo `Type`, que define el tipo de recurso de AWS que es. El atributo `Type` tiene el formato `AWS::ServiceName::ResourceType`. Por ejemplo, el atributo `Type` de un bucket de Amazon S3 es `AWS::S3::Bucket`.

Para obtener una lista completa de los tipos de recursos compatibles, consulte [Referencia de tipos de recursos y propiedades de AWS](#).

Propiedades de recursos

Las propiedades de recursos son opciones adicionales que puede especificar para definir los detalles de configuración para un tipo de recurso específico. Algunas propiedades son obligatorias, mientras que otras son opcionales. Algunas propiedades tienen valores predeterminados, por lo que especificarlas es opcional.

Para obtener información sobre las propiedades compatibles con cada tipo de recurso, consulte los temas en [Referencia de tipos de recursos y propiedades de AWS](#).

Los valores de la propiedad pueden ser cadenas literales, listas de cadenas, booleanos, referencias dinámicas, referencias de parámetros, pseudoreferencias o el valor devuelto por una función. En los siguientes ejemplos, se indica cómo declarar diferentes tipos de valores de propiedades:

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 físicos

Además de los IDs lógicos, determinados recursos también tienen un ID físico, que es el nombre asignado real de dicho recurso, como un ID de instancia de EC2 o un nombre de bucket de

S3. Utilice los ID físicos para identificar recursos fuera de las plantillas de CloudFormation, pero solo después de que se hayan creado los recursos. Por ejemplo, suponga que da a un recurso de instancia de EC2 un ID lógico de `MyEC2Instance`. Cuando CloudFormation crea la instancia, CloudFormation genera y asigna automáticamente un ID físico (como por ejemplo `i-1234567890abcdef0`) a la instancia. Puede utilizar este ID físico para identificar la instancia y ver sus propiedades (como, por ejemplo, el nombre de DNS) a través de la consola de Amazon EC2.

Para los buckets de Amazon S3 y muchos otros recursos, CloudFormation genera automáticamente un nombre físico único para el recurso si no se especifica uno de forma explícita. Este nombre físico se basa en una combinación del nombre de la pila de CloudFormation, el nombre lógico del recurso especificado en la plantilla de CloudFormation y un identificador único. Por ejemplo, si tiene un bucket de Amazon S3 con el nombre lógico `MyBucket` en una pila con nombre `MyStack`, CloudFormation podría dar nombre al bucket con el nombre físico `MyStack-MyBucket-abcdefghijkl1`.

En el caso de los recursos compatibles con los nombres personalizados, puede asignar sus propios nombres para identificar rápidamente los recursos. Por ejemplo, puede denominar un bucket de S3 que almacena registros `MyPerformanceLogs`. Para obtener más información, consulte [Tipo de nombre](#).

Recursos de referencia

A menudo, es necesario establecer las propiedades de un recurso en función del nombre o la propiedad de otro recurso. Por ejemplo, puede crear una instancia de EC2 que utilice los grupos de seguridad de EC2 o una distribución de CloudFront respaldada por un bucket de S3. Todos estos recursos se pueden crear en la misma plantilla de CloudFormation.

CloudFormation proporciona funciones intrínsecas que puede utilizar para hacer referencia a otros recursos y sus propiedades. Estas funciones le permiten crear dependencias entre los recursos y pasar valores de un recurso a otro.

La función de **Ref**

Por lo general, la función `Ref` se utiliza para recuperar una propiedad de identificación de los recursos definidos en la misma plantilla de CloudFormation. Los resultados dependen del tipo de recurso. En la mayoría de recursos, devuelve el nombre físico del recurso. Sin embargo, es posible que algunos tipos de recurso devuelvan un valor diferente, como, por ejemplo, una dirección IP para un recurso `AWS::EC2::EIP` o un Nombre de recurso de Amazon (ARN) para un tema de Amazon SNS.

Los siguientes ejemplos muestran cómo utilizar la función Ref en propiedades. En cada uno de estos ejemplos, el resultado de la función Ref será el nombre real del recurso LogicalResourceName declarado en otra parte de la plantilla. El ejemplo de sintaxis !Ref del ejemplo del archivo YAML es solo una forma más corta de escribir la función Ref.

JSON

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

YAML

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

Para obtener información detallada sobre el uso de la función Ref, consulte [Ref](#).

La función de Fn::GetAtt

La función Ref es útil si el parámetro o el valor que se brinda para un recurso es exactamente lo que desea. Sin embargo, es posible que necesite otros atributos de un recurso. Por ejemplo, si desea crear una distribución de CloudFront con un origen de S3, debe especificar la ubicación del bucket mediante una dirección de estilo DNS. Una serie de recursos tienen atributos adicionales cuyos valores puede utilizar en su plantilla. Para obtener estos atributos, utilice la función Fn::GetAtt.

Los siguientes ejemplos muestran cómo utilizar la función GetAtt en propiedades. La función Fn::GetAtt tiene dos parámetros: el nombre lógico del recurso y el nombre del atributo que debe recuperarse. El ejemplo de sintaxis !GetAtt del ejemplo del archivo YAML es solo una forma más corta de escribir la función GetAtt.

JSON

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

YAML

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

Para obtener información detallada sobre el uso de la función GetAtt, consulte [Fn::GetAtt](#).

Ejemplos

En los siguientes ejemplos, se ilustra cómo declarar los recursos y cómo las plantillas de CloudFormation pueden referenciar a otros recursos definidos en la misma plantilla y a los recursos de AWS existentes.

Temas

- [Declaración de un único recurso con un nombre personalizado](#)
- [Referencias a otros recursos con la función Ref](#)
- [Referencias a los atributos de los recursos con la función Fn::GetAtt](#)

Declaración de un único recurso con un nombre personalizado

En los siguientes ejemplos, se declara un único recurso de tipo `AWS::S3::Bucket` con el nombre lógico `MyBucket`. La propiedad `BucketName` está establecida en `amzn-s3-demo-bucket` y debe sustituirse por el nombre deseado para su bucket de S3.

Si utiliza esta declaración de recursos para crear una pila, CloudFormation creará un bucket de Amazon S3 con la configuración predeterminada. Para otros recursos, como una instancia de Amazon EC2 o un grupo de escalado automático, CloudFormation requiere más información.

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
```

Referencias a otros recursos con la función **Ref**

En los siguientes ejemplos, se muestra una declaración de recursos que define una instancia de EC2 y un grupo de seguridad. El recurso `Ec2Instance` referencia al recurso `InstanceSecurityGroup` como parte de su propiedad `SecurityGroupIds` con la función `Ref`. También incluye un grupo de seguridad existente (`sg-12a4c434`) que no está declarado en la plantilla. Debe utilizar cadenas literales para hacer referencia a recursos de AWS existentes.

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

```

Referencias a los atributos de los recursos con la función **Fn::GetAtt**

En los siguientes ejemplos, se muestra una declaración de recursos que define un recurso de distribución de CloudFront y un bucket de S3. El recurso `MyDistribution` especifica el nombre DNS del recurso `MyBucket` con la función `Fn::GetAtt` para obtener el atributo `DomainName` del bucket. Observará que la función `Fn::GetAtt` enumera sus dos parámetros en una matriz. Para funciones que utilizan varios parámetros, usted utiliza una matriz para especificar los parámetros.

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
```

Sintaxis de Parameters de la plantilla de CloudFormation

Utilice la sección `Parameters` opcional para personalizar sus plantillas. Con los parámetros, puede ingresar valores personalizados a su plantilla cada vez que crea o actualiza una pila. Al usar parámetros en sus plantillas, puede crear plantillas flexibles y reutilizables que se pueden adaptar a escenarios específicos.

Al definir los parámetros del tipo adecuado, puede elegir de una lista de identificadores de los recursos existentes cuando utilice la consola para crear su pila. Para obtener más información, consulte [Definición de recursos existentes en tiempo de ejecución con tipos de parámetros proporcionados por CloudFormation](#).

Los parámetros son una forma común de especificar los valores de las propiedades de los recursos de la pila. Sin embargo, es posible que haya ajustes que dependan de la región o sean algo complejos para que los usuarios los averigüen debido a otras condiciones o dependencias. En estos casos, debería incluir algo de lógica en la propia plantilla para que los usuarios puedan especificar valores más sencillos (o ninguno en absoluto) para obtener los resultados que desean. Por ejemplo, puede usar asignaciones. Para obtener más información, consulte [Sintaxis de Mappings de la plantilla de CloudFormation](#).

Sintaxis

Puede declarar parámetros en la sección `Parameters` de una plantilla, la cual usa la siguiente sintaxis general:

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
```

Un parámetro contiene una lista de atributos que definen su valor y limitaciones frente a su valor. El único atributo obligatorio es `Type`, que puede ser `String`, `Number` o un tipo de parámetro proporcionado por CloudFormation. También puede añadir un atributo `Description` que describe qué tipo de valor debe especificar. El nombre y la descripción del parámetro aparecen en la página [Especificar parámetros cuando utiliza la plantilla en el asistente de Crear pila](#).

Note

De forma predeterminada, la consola de CloudFormation ordena alfabéticamente parámetros de entrada por su ID lógico. Para anular este orden predeterminado y agrupar los parámetros relacionados, puede usar la clave de metadatos `AWS::CloudFormation::Interface` de su plantilla. Para obtener más información, consulte [Organización de los parámetros de CloudFormation con metadatos AWS::CloudFormation::Interface](#).

Para parámetros con valores predeterminados, CloudFormation utiliza los valores predeterminados a menos que los usuarios especifiquen otro valor. Si omite el atributo predeterminado, los usuarios deben especificar un valor para ese parámetro. Sin embargo, exigir que el usuario ingrese un

valor no garantiza que el valor sea válido. Para validar el valor de un parámetro, puede declarar limitaciones o indicar un tipo de parámetro específico de AWS.

En el caso de parámetros sin valores predeterminados, los usuarios deben especificar un valor de nombre de clave al crear la pila. Caso contrario, CloudFormation no logra crear la pila y arroja una excepción:

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

Propiedades

AllowedPattern

Una expresión regular que representa los patrones que hay que permitir para tipos `String` o `CommaDelimitedList`. Cuando se aplica a un parámetro del tipo `String`, el patrón debe coincidir con todo el valor del parámetro proporcionado. Cuando se aplica a un parámetro del tipo `CommaDelimitedList`, el patrón debe coincidir con cada valor de la lista.

Obligatorio: no

AllowedValues

Una matriz que contiene la lista de valores permitidos para el parámetro. Cuando se aplica a un parámetro del tipo `String`, el valor del parámetro debe ser uno de los valores permitidos. Cuando se aplica a un parámetro del tipo `CommaDelimitedList`, cada valor de la lista debe ser uno de los valores permitidos especificados.

Obligatorio: no

Note

Si usa YAML y quiere usar cadenas `Yes` y `No` para `AllowedValues`, use comillas simples para evitar que el analizador YAML considere estos valores booleanos.

ConstraintDescription

Una cadena que explica una restricción cuando se infringe la restricción. Por ejemplo, sin una descripción de la restricción, un parámetro con un patrón permitido de `[A-Za-z0-9]+` muestra el siguiente mensaje de error cuando el usuario especifica un valor no válido:

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

Al añadir una descripción de restricción, como, por ejemplo debe contener únicamente letras (mayúsculas y minúsculas) y números, puede mostrar el siguiente mensaje de error personalizado:

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

Obligatorio: no

Default

Un valor del tipo apropiado para la plantilla que se debe utilizar si no se especifica ningún valor al crear una pila. Si define restricciones para el parámetro, debe especificar un valor que cumpla dichas restricciones.

Obligatorio: no

Description

Una cadena de hasta 4000 caracteres que describe el parámetro.

Obligatorio: no

MaxLength

Cualquier valor entero que determina el mayor número de caracteres que desea permitir para tipos de `String`.

Obligatorio: no

MaxValue

Un valor numérico que determina el mayor valor numérico que desea permitir para tipos de `Number`.

Obligatorio: no

MinLength

Un valor entero que determina el menor número de caracteres que desea permitir para tipos de `String`.

Obligatorio: no

MinValue

Un valor numérico que determina el menor valor numérico que desea permitir para tipos de `Number`.

Obligatorio: no

NoEcho

Si se va a enmascarar el valor del parámetro para impedir que se muestre en la consola, en las herramientas de línea de comandos o en la API. Si establece el atributo `NoEcho` en `true`, CloudFormation devuelve el valor del parámetro enmascarado como asteriscos (`*****`) para cualquier llamada que describa la pila o los eventos de la pila, excepto para la información almacenada en las ubicaciones especificadas a continuación.

Obligatorio: no

Important

El uso del atributo `NoEcho` no enmascara ninguna información almacenada en lo que se muestra a continuación:

- La sección de la plantilla `Metadata`. CloudFormation no transforma, modifica ni redacta ninguna información que incluya en la sección `Metadata`. Para obtener más información, consulte [Metadata](#).
- La sección de la plantilla `Outputs`. Para obtener más información, consulte [Salidas](#).
- El atributo `Metadata` de una definición de recurso. Para obtener más información, consulte [Metadata atributo](#).

Recomendamos encarecidamente que no utilice estos mecanismos para incluir información confidencial, como contraseñas o secretos.

Important

En lugar de integrar información confidencial directamente en las plantillas de CloudFormation, se recomienda utilizar parámetros dinámicos en la plantilla de la pila para hacer referencia a la información confidencial almacenada y administrada fuera de CloudFormation, como en AWS Systems Manager Parameter Store o AWS Secrets Manager.

Para obtener más información, consulte la práctica recomendada [No integre credenciales en sus plantillas](#).

 Important

Recomendamos encarecidamente no incluir parámetros NoEcho ni ningún dato confidencial en las propiedades de los recursos que formen parte del identificador principal de un recurso.

Cuando se incluye un parámetro de NoEcho en una propiedad que forma un identificador de recursos principal, CloudFormation puede usar el valor de texto sin formato real en el identificador del recurso principal. Este ID de recurso puede aparecer en cualquier destino o salida derivada.

Para determinar qué propiedades de recursos comprenden el identificador principal de un tipo de recurso, consulte la documentación de referencia de recursos para ese recurso en la [Referencia de tipos de recursos y propiedades de AWS](#). En la sección Return values (Valores de devolución), el valor de devolución de la función Ref representa las propiedades del recurso que comprenden el identificador principal del tipo de recurso.

Type

El tipo de datos para el parámetro (DataType).

Obligatorio: sí

CloudFormation es compatible con los siguientes tipos de parámetros:

String

Una cadena literal. Puede usar los siguientes atributos para declarar las limitaciones: MinLength, MaxLength, Default, AllowedValues y AllowedPattern.

Por ejemplo, los usuarios podrían especificar "MyUserName".

Number

Un número entero o float. CloudFormation valida el valor del parámetro como un número; sin embargo, cuando utiliza el parámetro en otro sitio en la plantilla (por ejemplo, mediante el uso de la función intrínseca Ref), el valor del parámetro se convierte en una cadena.

Puede usar los siguientes atributos para declarar las limitaciones: `MinValue`, `MaxValue`, `Default` y `AllowedValues`.

Por ejemplo, los usuarios podrían especificar `"8888"`.

`List<Number>`

Una matriz de valores enteros o floats que están separados por comas. CloudFormation valida el valor del parámetro como números; sin embargo, cuando utiliza el parámetro en otro sitio en la plantilla (por ejemplo, mediante el uso de la función intrínseca `Ref`), el valor del parámetro se convierte en una lista de cadenas.

Por ejemplo, los usuarios podrían especificar `"80, 20"` y una `Ref` daría lugar a `["80", "20"]`.

`CommaDelimitedList`

Una matriz de cadenas literales que están separadas por comas. El número total de cadenas debería ser uno más que el número total de comas. Además, en cada cadena de miembro se elimina el espacio.

Por ejemplo, los usuarios podrían especificar `"test, dev, prod"` y una `Ref` daría lugar a `["test", "dev", "prod"]`.

AWSTipos de parámetros específicos de

Valores de AWS, como nombres de pares de claves e ID de VPC de Amazon EC2. Para obtener más información, consulte [Definición de recursos existentes en tiempo de ejecución](#).

Tipos de parámetros de Systems Manager

Parámetros correspondientes a los parámetros existentes en Administrador de sistemas Parameter Store. Puede especificar una clave de parámetro de Systems Manager como valor del tipo de parámetro de Systems Manager, y CloudFormation recupera el valor más reciente del Almacén de parámetros que se usará para la pila. Para obtener más información, consulte [Definición de recursos existentes en tiempo de ejecución](#).

Requisitos generales para los parámetros

Los siguientes requisitos se aplican al usar los parámetros:

- Puede tener un máximo de 200 parámetros en una plantilla de CloudFormation.

- Cada parámetro debe recibir un nombre lógico (también llamado ID lógico), que tiene que ser alfanumérico y único entre todos los nombres lógicos dentro de la plantilla.
- A cada parámetro se le debe asignar un tipo de parámetro compatible con CloudFormation. Para obtener más información, consulte [Tipo](#).
- A cada parámetro se le debe asignar un valor en tiempo de ejecución para que CloudFormation aprovisiona correctamente la pila. Si lo desea, puede especificar un valor predeterminado para que CloudFormation lo use a menos que se proporcione otro valor.
- Los parámetros se deben declarar y se debe hacer referencia a ellos desde la misma plantilla. Puede hacer referencia a los parámetros en las secciones Resources y Outputs de la plantilla.

Ejemplos

Temas

- [Parámetro de cadena simple](#)
- [Parámetro de contraseña](#)
- [Parámetros de referencia](#)
- [Parámetros de lista delimitada por comas](#)
- [Devuelva un valor de un parámetro de lista delimitada por comas](#)

Parámetro de cadena simple

En el siguiente ejemplo se declara un parámetro denominado `InstanceTypeParameter` de tipo `String`. Este parámetro le permite especificar el tipo de instancia de Amazon EC2 para la pila. Si no se proporciona ningún valor durante la creación o actualización de la pila, CloudFormation utiliza el valor predeterminado de `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
```

Parámetro de contraseña

En el siguiente ejemplo se declara un parámetro denominado DBPwd de tipo `String` con ningún valor predeterminado. La propiedad `NoEcho` está configurada en `true` para evitar que el valor del parámetro se muestre en las descripciones de las pilas. La longitud mínima que se puede especificar es 1 y la longitud máxima que se puede especificar es 41. El patrón permite caracteres alfabéticos en mayúsculas y minúsculas, y números. Este ejemplo también ilustra el uso de una expresión regular para la propiedad `AllowedPattern`.

JSON

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

YAML

```
Parameters:
  DBPwd:
    NoEcho: true
    Description: The database admin account password
    Type: String
    MinLength: 1
```

```
MaxLength: 41
AllowedPattern: ^[a-zA-Z0-9]*$
```

Parámetros de referencia

Puede usar la función intrínseca `Ref` para hacer referencia a un parámetro, mientras que CloudFormation utiliza el valor del parámetro para aprovisionar la pila. Puede hacer referencia a los parámetros en las secciones `Resources` y `Outputs` de la misma plantilla.

En el siguiente ejemplo, la propiedad `InstanceType` del recurso de instancia de EC2 hace referencia al valor del parámetro `InstanceTypeParameter`:

JSON

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

YAML

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

Parámetros de lista delimitada por comas

El tipo de parámetro `CommaDelimitedList` puede resultar útil cuando se necesitan proporcionar varios valores para una sola propiedad. El siguiente ejemplo declara un parámetro denominado `DbSubnetIpBlocks` con un valor predeterminado de tres bloques CIDR separados por comas.

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"

```

Devuelva un valor de un parámetro de lista delimitada por comas

Para consultar un valor específico en una lista delimitada por comas del parámetro, utilice la función intrínseca `Fn::Select` en la sección `Resources` de la plantilla. Transfiera el valor de índice del objeto que desea y una lista de objetos, tal y como se muestra en el siguiente ejemplo.

JSON

```

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

```

```

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

```

```

    "Type": "AWS::EC2::Subnet",
    "Properties": {
      "AvailabilityZone": {
        "Fn::Sub": [
          "${AWS::Region}${AZ}",
          {
            "AZ": {
              "Fn::Select": [
                2,
                { "Ref": "VpcAzs" }
              ]
            }
          }
        ],
      },
      "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: !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
```

Recursos relacionados

CloudFormation también admite el uso de referencias dinámicas para especificar valores de propiedades de forma dinámica. Por ejemplo, puede que necesite hacer referencia a cadenas seguras almacenadas en el almacén de parámetros de Systems Manager. Para obtener más información, consulte [Obtención de valores almacenados en otros servicios con referencias dinámicas](#).

También puede utilizar pseudoparámetros dentro de una función Ref o Sub para rellenar valores de forma dinámica. Para obtener más información, consulte [Obtención de valores de AWS mediante pseudoparámetros](#).

Sintaxis de Outputs de la plantilla de CloudFormation

La sección opcional `Outputs` declara los valores de salida de la pila. Estos valores de salida se pueden usar de distintas maneras:

- Captura de detalles importantes sobre sus recursos: una salida es una manera cómoda de capturar información importante sobre sus recursos. Por ejemplo, puede declarar la salida para el nombre del bucket de S3 para una pila para que sea más fácil encontrar el bucket. Puede ver los valores de salida en la pestaña Salidas de la consola de CloudFormation o con el comando [describe-stacks](#) de la CLI.
- Referencias entre pilas: puede importar valores de salida a otras pilas para [crear referencias entre pilas](#). Esto es útil cuando necesita compartir recursos o configuraciones entre varias pilas.

Important

CloudFormation no redacta ni enmascara ninguna información que incluya en la sección `Outputs`. Recomendamos encarecidamente que no utilice esta sección para generar información confidencial, como contraseñas o secretos.

Los valores de salida están disponibles una vez que la operación de pila está completa.

Los valores de salida de la pila no están disponibles cuando el estado de la pila está en cualquiera de estos [estados](#) `IN_PROGRESS`. No recomendamos establecer dependencias entre un tiempo de ejecución de servicio y el valor de salida de la pila porque los valores de salida pueden no estar disponibles en todo momento.

Sintaxis

La sección `Outputs` consta del nombre de clave `Outputs`. Puede declarar un máximo de 200 salidas en una plantilla.

El siguiente ejemplo muestra la estructura de la sección `Outputs`.

JSON

Utilice llaves para delimitar todas las declaraciones de salida. Delimite varias salidas con comas.


```
"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
```

Campos de salida

La sección `Outputs` puede incluir los siguientes campos.

ID lógico (también denominado nombre lógico)

Un identificador para la salida actual. El ID lógico tiene que ser alfanumérico (a–z, A–Z, 0–9) y único dentro de la plantilla.

Description (opcional)

Un tipo de `String` que describe el valor de salida. El valor de la declaración de la descripción debe ser una cadena literal de entre 0 y 1024 bytes de longitud. No puede utilizar un parámetro o función para especificar la descripción.

Value (obligatorio)

El valor de la propiedad devuelto por el comando [describe-stacks](#). El valor de una salida puede incluir literales, referencias de parámetros, pseudoparámetros, un valor de asignación o funciones intrínsecas.

Export (opcional)

El nombre de la salida de los recursos que se exportan para una referencia cruzada de pila.

Puede utilizar funciones intrínsecas para personalizar el valor Name de una exportación.

Para obtener más información, consulte [Obtención de salidas exportadas de una pila implementada de CloudFormation](#).

Para asociar una condición a una salida, defina la condición en la sección [Conditions](#) de la plantilla.

Ejemplos

Los siguientes ejemplos ilustran cómo funciona la salida de la pila.

Temas

- [Salida de la pila](#)
- [Personalización del nombre de exportación mediante Fn::Sub](#)
- [Personalización del nombre de exportación mediante Fn::Join](#)
- [Devuelve una URL creada con Fn::Join](#)

Salida de la pila

En el siguiente ejemplo, la salida denominada BackupLoadBalancerDNSName devuelve el nombre de DNS para el recurso con el ID lógico BackupLoadBalancer solo cuando la condición CreateProdResources es true. La salida con el nombre InstanceID devuelve el ID de la instancia de EC2 con el ID lógico EC2Instance.

JSON

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

YAML

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

Personalización del nombre de exportación mediante **Fn::Sub**

En los siguientes ejemplos, la salida denominada `StackVPC` devuelve el ID de una VPC y, a continuación, exporta el valor para la referencia cruzada de la pila `VPCID` junto al nombre de la pila.

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"
```

Para obtener más información sobre la función `Fn::Sub`, consulte [Fn::Sub](#).

Personalización del nombre de exportación mediante **Fn::Join**

También puede usar de la función `Fn::Join` para crear valores basados en parámetros, atributos de recursos y otras cadenas.

En los ejemplos siguientes, se utiliza la función `Fn::Join` para personalizar el nombre de la exportación en lugar de la función `Fn::Sub`. La función `Fn::Join` de ejemplo concatena el nombre de la pila con el nombre VPCID utilizando dos puntos como separador.

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 ] ]
```

Para obtener más información sobre la función `Fn::Join`, consulte [Fn::Join](#).

Devuelve una URL creada con **Fn::Join**

En el siguiente ejemplo de una plantilla para crear un sitio de WordPress, `InstallURL` es la cadena devuelta por una llamada a una función `Fn::Join` que concatena `http://`, el nombre DNS del recurso `ElasticLoadBalancer` y `/wp-admin/install.php`. El valor de salida debería ser similar al siguiente:

```
http://mywptests-elasticl-1gb5116sl8y5v-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
```

Para obtener más información sobre la función `Fn::Join`, consulte [Fn::Join](#).

Sintaxis de Mappings de la plantilla de CloudFormation

La sección opcional `Mappings` ayuda a crear pares de clave-valor que pueden usarse para especificar valores a ciertas condiciones o dependencias.

Un caso de uso común es para la sección `Mappings` es para establecer valores según la Región de AWS en la que la pila esté implementada. Esto se puede lograr con el pseudoparámetro `AWS::Region`. El pseudoparámetro `AWS::Region` es un valor que CloudFormation resuelve para la región en la que se crea la pila. CloudFormation resuelve los pseudoparámetros cuando usted crea la pila.

Para recuperar los valores de una asignación, puede usar la función intrínseca `Fn::FindInMap` en la sección `Resources` de la plantilla.

Sintaxis

La sección `Mappings` usa la siguiente sintaxis:

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 es el nombre lógico para la asignación.
- Dentro de las asignaciones, cada una es una clave seguida de otra asignación.
- La clave debe ser un mapa de pares de nombre-valor y única dentro del mapeo.
- El par nombre-valor es una etiqueta y el valor que se deben asignar. Al nombrar los valores, puede asignar más de un conjunto de valores a una clave.
- Las claves en mapeos deben ser cadenas literales.
- Los valores pueden ser de tipo String o List.

Note

No puede incluir parámetros, pseudo parámetros o funciones intrínsecas en la sección Mappings.

Si actualmente su pila no utiliza los valores de una asignación, no puede actualizar solo la asignación. Tiene que incluir cambios que agreguen, modifiquen o eliminen recursos.

Ejemplos

Temas

- [Asignación básica](#)
- [Asignación con múltiples valores](#)
- [Devolución de un valor a partir de una asignación](#)
- [Parámetro de entrada y Fn::FindInMap](#)

Asignación básica

El siguiente ejemplo muestra la sección Mappings con una asignación RegionToInstanceType, que contiene cinco claves que se asignan a pares nombre-valor que contienen valores de una única cadena. Las claves son nombres de regiones. Cada par nombre-valor es un tipo de instancia de la familia T que está disponible en la región representada por la clave. Los pares nombre-valor tienen un nombre (InstanceType en el ejemplo) y un valor.

JSON

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

YAML

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

Asignación con múltiples valores

En el siguiente ejemplo, se muestran claves de región que se asignan a dos conjuntos de valores: uno denominado MyAMI1 y el otro, MyAMI2.

Note

Los ID de AMI que se muestran en estos ejemplos son marcadores de posición con fines de demostración. Siempre que sea posible, considere la posibilidad de utilizar referencias dinámicas a los parámetros AWS Systems Manager como alternativa a la sección `Mappings`. Para evitar actualizar todas las plantillas con un nuevo ID cada vez que cambie la AMI que desea utilizar, utilice un parámetro AWS Systems Manager para obtener el último ID de AMI al crear o actualizar la pila. Las versiones más recientes de las AMI más utilizadas también están disponibles como parámetros públicos en Systems Manager. Para

obtener más información, consulte [Obtención de valores almacenados en otros servicios con referencias dinámicas](#).

JSON

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

YAML

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

Devolución de un valor a partir de una asignación

Puede utilizar la función `Fn::FindInMap` para obtener un valor identificado a partir de una clave especificada. La siguiente plantilla de ejemplo contiene un recurso de Amazon EC2 cuya propiedad `InstanceType` asigna la función `FindInMap`. La función `FindInMap` especifica clave como la región de Región de AWS donde se crea la pila (mediante el pseudoparámetro `AWS::Region`) y `InstanceType` como el nombre del valor al que hay que asignarla. `ImageId` utiliza un parámetro de Systems Manager para recuperar de forma dinámica la AMI más reciente de Amazon Linux 2. Para obtener más información sobre pseudoparámetros, consulte . [Obtención de valores de AWS mediante pseudoparámetros](#).

JSON

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Mappings" : {
    "RegionToInstanceType" : {
      "us-east-1"      : { "InstanceType" : "t2.micro" },
      "us-west-1"      : { "InstanceType" : "t2.micro" },
      "eu-west-1"      : { "InstanceType" : "t2.micro" },
      "eu-north-1"     : { "InstanceType" : "t3.micro" },
      "me-south-1"     : { "InstanceType" : "t3.micro" }
    }
  },
  "Resources" : {
    "myEC2Instance" : {
      "Type" : "AWS::EC2::Instance",
      "Properties" : {
        "ImageId" : "{resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2}",
        "InstanceType" : { "Fn::FindInMap" : [ "RegionToInstanceType", { "Ref" :
"AWS::Region" }, "InstanceType" ] }
      }
    }
  }
}
```

YAML

```
AWSTemplateFormatVersion: 2010-09-09
Mappings:
  RegionToInstanceType:
```

```

us-east-1:
  InstanceType: t2.micro
us-west-1:
  InstanceType: t2.micro
eu-west-1:
  InstanceType: t2.micro
eu-north-1:
  InstanceType: t3.micro
me-south-1:
  InstanceType: t3.micro
Resources:
  myEC2Instance:
    Type: AWS::EC2::Instance
    Properties:
      ImageId: '{{resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-
x86_64-gp2}}'
      InstanceType: !FindInMap [RegionToInstanceType, !Ref 'AWS::Region', InstanceType]

```

Parámetro de entrada y **Fn::FindInMap**

La siguiente plantilla de ejemplo muestra cómo crear una instancia de EC2 con varias asignaciones. La plantilla utiliza asignaciones anidadas para seleccionar automáticamente el tipo de instancia y el grupo de seguridad adecuados en función de la Región de AWS de destino y el tipo de entorno (Dev o Prod). También utiliza un parámetro de Systems Manager para recuperar de forma dinámica la AMI más reciente de Amazon Linux 2.

JSON

```

{
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Parameters" : {
    "EnvironmentType" : {
      "Description" : "The environment type (Dev or Prod)",
      "Type" : "String",
      "Default" : "Dev",
      "AllowedValues" : [ "Dev", "Prod" ]
    }
  },
  "Mappings" : {
    "RegionAndEnvironmentToInstanceType" : {
      "us-east-1"      : { "Dev" : "t3.micro", "Prod" : "c5.large" },
      "us-west-1"      : { "Dev" : "t2.micro", "Prod" : "m5.large" }
    }
  },

```

```

    "RegionAndEnvironmentToSecurityGroup" : {
      "us-east-1"      : { "Dev" : "sg-12345678", "Prod" : "sg-abcdef01" },
      "us-west-1"     : { "Dev" : "sg-ghijkl23", "Prod" : "sg-45678abc" }
    }
  },
  "Resources" : {
    "Ec2Instance" : {
      "Type" : "AWS::EC2::Instance",
      "Properties" : {
        "ImageId" : "{{resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2}}",
        "InstanceType" : { "Fn::FindInMap": [ "RegionAndEnvironmentToInstanceType",
        { "Ref": "AWS::Region" }, { "Ref": "EnvironmentType" } ] },
        "SecurityGroupIds" : [{ "Fn::FindInMap" :
        [ "RegionAndEnvironmentToSecurityGroup", { "Ref" : "AWS::Region" }, { "Ref" :
        "EnvironmentType" } ]}]
      }
    }
  }
}

```

YAML

```

AWSTemplateFormatVersion: 2010-09-09
Parameters:
  EnvironmentType:
    Description: The environment type (Dev or Prod)
    Type: String
    Default: Dev
    AllowedValues:
      - Dev
      - Prod
Mappings:
  RegionAndEnvironmentToInstanceType:
    us-east-1:
      Dev: t3.micro
      Prod: c5.large
    us-west-1:
      Dev: t2.micro
      Prod: m5.large
  RegionAndEnvironmentToSecurityGroup:
    us-east-1:
      Dev: sg-12345678

```

```
    Prod: sg-abcdef01
  us-west-1:
    Dev: sg-ghijkl23
    Prod: sg-45678abc
Resources:
  Ec2Instance:
    Type: AWS::EC2::Instance
    Properties:
      ImageId: '{{resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-
x86_64-gp2}}'
      InstanceType: !FindInMap [RegionAndEnvironmentToInstanceType, !Ref
'AWS::Region', !Ref EnvironmentType]
      SecurityGroupIds:
        - !FindInMap [RegionAndEnvironmentToSecurityGroup, !Ref 'AWS::Region', !Ref
EnvironmentType]
```

Recursos relacionados

Estos temas relacionados pueden resultarle útiles a la hora de desarrollar plantillas que utilicen la función `Fn::FindInMap`.

- [Fn::FindInMap](#)
- [Fn::FindInMap Mejoras de](#)
- [Fn::Sub](#)

Sintaxis de Metadata de la plantilla de CloudFormation

Metadata almacena información adicional mediante objetos JSON o YAML. Los tipos de metadatos a nivel de plantilla que puede usar en su plantilla incluyen lo siguiente:

Metadatos personalizados

Almacena pares clave-valor definidos por el usuario. Por ejemplo, puede proporcionar información adicional que no afecte a la creación de recursos, pero que ofrezca un contexto adicional sobre las características específicas de la infraestructura, el equipo o la implementación.

`AWS::CloudFormation::Interface`

Define el agrupamiento y ordenación de parámetros de entrada cuando se muestran en la consola de CloudFormation. De forma predeterminada, la consola de CloudFormation ordena alfabéticamente parámetros por su ID lógico.

AWS::CloudFormation::Designer

CloudFormation Designer (Designer) alcanzó el fin de su vida útil el 5 de febrero de 2025.

Important

Durante la actualización de una pila, no puede actualizar la sección Metadata por sí misma. Puede actualizarla solo cuando incluye cambios que añaden, modifican o eliminan recursos. CloudFormation no transforma, modifica ni redacta ninguna información que incluya en la sección Metadata. Debido a esto, recomendamos encarecidamente que no utilice esta sección para almacenar información confidencial, como contraseñas o secretos.

Sintaxis

Para declarar estos metadatos personalizados en su plantilla de CloudFormation, use la siguiente sintaxis:

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"
```

Para ver la sintaxis de `AWS::CloudFormation::Interface`, consulte [Organización de los parámetros de CloudFormation con metadatos AWS::CloudFormation::Interface](#).

Organización de los parámetros de CloudFormation con metadatos

AWS::CloudFormation::Interface

`AWS::CloudFormation::Interface` es una clave de metadatos que define cómo se agrupan y ordenan los parámetros en la consola de CloudFormation. De forma predeterminada, cuando crea o actualiza pilas en la consola, la consola muestra parámetros de entrada en orden alfabético por su ID lógico. Al utilizar esta clave, puede definir su propia agrupación y orden de parámetros para que los usuarios puedan especificar los valores de los parámetros de manera eficiente. Por ejemplo, puede agrupar todos los parámetros relacionados con EC2 en un grupo y todos los parámetros relacionados con la VPC en otro grupo.

En la clave de metadatos, puede especificar los grupos que desea crear, los parámetros que desea incluir en cada grupo y el orden en el que la consola muestra cada parámetro dentro de su grupo.

También puede definir etiquetas para los parámetros. Una etiqueta es una descripción o nombre fáciles de recordar que muestra la consola en lugar de un ID lógico del parámetro. Las etiquetas son útiles para ayudar a los usuarios a entender los valores que se deben especificar para cada parámetro. Por ejemplo, puede etiquetar un parámetro `KeyPair Select` an `EC2 key pair`.

Todos los parámetros a los que se hace referencia en la clave de metadatos deben declararse en la sección `Parameters` de la plantilla.

Note

Solo la consola de CloudFormation utiliza la clave de metadatos `AWS::CloudFormation::Interface`. AWS CLI y llamadas a la API no utilizan esta clave.

Sintaxis

Para declarar esta entidad en su plantilla de CloudFormation, use la siguiente sintaxis:

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

```

Propiedades

ParameterGroups

Una lista de tipos de grupos de parámetros, donde debe especificar los nombres de los grupos, los parámetros de cada grupo y el orden en el que se muestran los parámetros.

Obligatorio: no

Label

Un nombre para el grupo de parámetros.

Obligatorio: no

default

La etiqueta predeterminada que la consola CloudFormation utiliza para denominar un grupo de parámetro.

Obligatorio: no

Tipo: cadena

Parameters

Una lista de IDs lógicos de parámetros sensibles a mayúsculas y minúsculas que se incluirán en el grupo. Los parámetros ya deben estar definidos en la sección `Parameters` de la plantilla. Se puede incluir un parámetro en un único grupo de parámetros.

La consola muestra los parámetros que no asocia a un grupo de parámetros por orden alfabético en el grupo `Other parameters`.

Obligatorio: no

Tipo: lista de valores de cadena

ParameterLabels

Una asignación de parámetros y sus nombres fáciles de recordar que muestra la consola de CloudFormation cuando se crea o actualiza una pila.

Obligatorio: no

Etiqueta de parámetro

Una etiqueta para un parámetro. La etiqueta define una descripción o un nombre fácil de recordar que la consola de CloudFormation muestra en la página `Specify Parameters` (Especificar parámetros) cuando se crea o actualiza una pila. La etiqueta de parámetro debe ser el ID lógico que distingue entre mayúsculas y minúsculas de un parámetro válido declarado en la sección `Parameters` de la plantilla.

Obligatorio: no

default

La etiqueta predeterminada que la consola de CloudFormation utiliza para denominar un parámetro.

Obligatorio: no

Tipo: cadena

Ejemplo

El siguiente ejemplo define dos grupos de parámetros: Network Configuration y Amazon EC2 Configuration. El grupo Network Configuration incluye los parámetros VPCID, SubnetId y SecurityGroupId que se definen en la sección Parameters de la plantilla (no se muestra). El orden en el que la consola muestra estos parámetros se define por el orden en el que se enumeran los parámetros, empezando por el parámetro VPCID. El ejemplo agrupa y ordena de manera parecida los parámetros Amazon EC2 Configuration.

El ejemplo también define una etiqueta para el parámetro VPCID. La consola mostrará Which VPC should this be deployed to? (¿En qué VPC debe implementarse esto?) en lugar del ID lógico del parámetro (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?" }
    }
  }
}
```

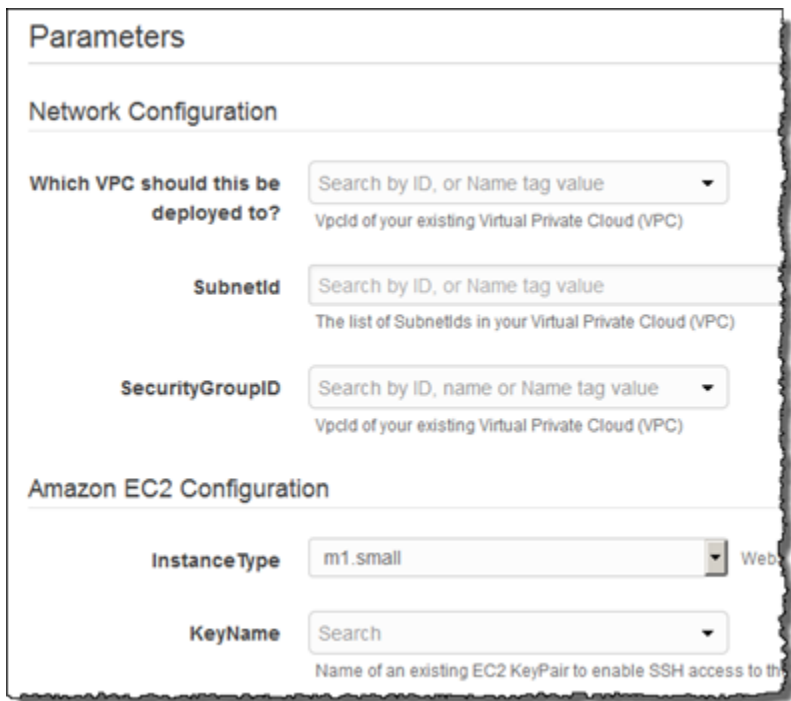
YAML

```
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?"
```

Grupos de parámetros en la consola

Con la clave de metadatos de este ejemplo, el siguiente gráfico ilustra cómo muestra la consola grupos de parámetros cuando se crea o actualiza una pila: Grupos de parámetros en la consola



The screenshot shows the 'Parameters' section of the AWS CloudFormation console. It is divided into two sections: 'Network Configuration' and 'Amazon EC2 Configuration'.

Network Configuration

- Which VPC should this be deployed to?**: A dropdown menu with the placeholder text 'Search by ID, or Name tag value'. Below it, a note says 'Vpcid of your existing Virtual Private Cloud (VPC)'.
- SubnetId**: A dropdown menu with the placeholder text 'Search by ID, or Name tag value'. Below it, a note says 'The list of SubnetIds in your Virtual Private Cloud (VPC)'.
- SecurityGroupID**: A dropdown menu with the placeholder text 'Search by ID, name or Name tag value'. Below it, a note says 'Vpcid of your existing Virtual Private Cloud (VPC)'.

Amazon EC2 Configuration

- InstanceType**: A dropdown menu with 'm1.small' selected. To its right, the text 'Web' is visible.
- KeyName**: A dropdown menu with the placeholder text 'Search'. Below it, a note says 'Name of an existing EC2 KeyPair to enable SSH access to the instance'.

Sintaxis de Rules de la plantilla de CloudFormation

La sección `Rules` es una parte opcional de una plantilla de CloudFormation que permite una lógica de validación personalizada. Cuando se incluye, esta sección contiene funciones de reglas que

validan los valores de los parámetros antes de que CloudFormation cree o actualice cualquier recurso.

Las reglas son útiles cuando las restricciones estándar de los parámetros son insuficientes. Por ejemplo, cuando SSL está habilitado, se debe proporcionar tanto un certificado como un nombre de dominio. Una regla puede garantizar que se cumplan estas dependencias.

Sintaxis

La sección `Rules` usa la siguiente sintaxis:

JSON

La sección `Rules` de una plantilla consta del nombre de clave `Rules`, seguido de un único signo de dos puntos. Debe utilizar llaves para incluir todas las declaraciones de reglas. Si se declaran varias reglas, deben delimitarse mediante comas. Para cada regla, se declara un nombre lógico entre comillas seguido de un signo de dos puntos y de las llaves que contienen la condición de regla y las declaraciones.

```
{
  "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

```

Campos de reglas

La sección **Rules** puede incluir los siguientes campos.

ID lógico (también denominado nombre lógico)

Un identificador único para cada regla.

RuleCondition (opcional)

Una propiedad que determina cuándo surte efecto una regla. Si no se define la condición de regla, las declaraciones de la regla surten efecto en todos los casos. Para cada regla, sólo puede definir una condición de regla.

Assertions (obligatorio)

Una o más declaraciones que especifican los valores aceptables para un parámetro determinado.

Assert

Una condición que se debe evaluar como `true`.

AssertDescription

Un mensaje que se muestra cuando falla una aserción.

Funciones intrínsecas específicas de la regla

Para definir sus reglas, debe utilizar funciones específicas de reglas, que son funciones que solo se pueden utilizar en la sección `Rules` de una plantilla. Puede anidar funciones, pero el resultado final de una condición de regla o una declaración debe ser `true` o `false`.

Ahora están disponibles las siguientes funciones de reglas:

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

Estas funciones intrínsecas se utilizan en la condición o aserciones de una regla. La propiedad de condición determina si CloudFormation aplicará las declaraciones. Si la condición se evalúa como `true`, CloudFormation evalúa las declaraciones para comprobar si el valor de un parámetro es válido en el momento de crear o actualizar un producto provisionado. Si el valor de un parámetro no es válido, CloudFormation no crea o actualizar la pila. Si la condición se evalúa como `false`,

CloudFormation no comprueba el valor del parámetro y procede a llevar a cabo la operación de la pila.

Ejemplos

Temas

- [Comprobación condicional del valor de un parámetro](#)
- [Validación de parámetros cruzados](#)

Comprobación condicional del valor de un parámetro

En el siguiente ejemplo, las dos reglas comprueban el valor del InstanceType parámetro. En función del valor del parámetro entorno (test o prod), el usuario debe especificar t3.medium o t3.large para el parámetro InstanceType. Los parámetros InstanceType y Environment deben declararse en la sección Parameters de la misma plantilla.

JSON

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

```

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

```

YAML

```

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

```



```

- - t3.large
- !Ref InstanceType
  AssertDescription: 'For a production environment, the instance type must be
t3.large'

```

Validación de parámetros cruzados

Las siguientes plantillas de ejemplo demuestran el uso de reglas para las validaciones entre parámetros. Crean un sitio web de ejemplo que se ejecuta en un grupo de escalado automático detrás de un equilibrador de carga. El sitio web está disponible en los puertos 80 o 443, según los parámetros de entrada. Las instancias del grupo de escalado automático se pueden configurar para que escuchen en cualquier puerto (con 8888 como el valor predeterminado).

Las reglas de esta plantilla validan los parámetros de entrada antes de crear la pila. Comprueban que todas las subredes pertenecen a la VPC especificada y garantizan que, cuando el parámetro UseSSL se establece en Yes, se proporcionen un ARN de certificado SSL y un nombre de zona alojada.

Note

Se le facturarán los recursos de AWS que utilice si crea una pila a partir de esta plantilla.

JSON

```

{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Parameters": {
    "VpcId": {
      "Type": "AWS::EC2::VPC::Id",
      "Description": "VpcId of your existing Virtual Private Cloud (VPC)",
      "ConstraintDescription": "must be the VPC Id of an existing Virtual Private
Cloud."
    },
    "Subnets": {
      "Type": "List<AWS::EC2::Subnet::Id>",
      "Description": "The list of SubnetIds in your Virtual Private Cloud (VPC)",
      "ConstraintDescription": "must be a list of at least two existing subnets
associated with at least two different availability zones."
    },
    "InstanceType": {
      "Description": "WebServer EC2 instance type",

```

```

    "Type": "String",
    "Default": "t2.micro",
    "AllowedValues": ["t2.micro", "t3.micro"],
    "ConstraintDescription": "must be a valid EC2 instance type."
  },
  "KeyName": {
    "Description": "Name of an existing EC2 KeyPair to enable SSH access to the
instances",
    "Type": "AWS::EC2::KeyPair::KeyName",
    "ConstraintDescription": "must be the name of an existing EC2 KeyPair."
  },
  "SSHLocation": {
    "Description": "The IP address range that can be used to SSH to the EC2
instances",
    "Type": "String",
    "MinLength": "9",
    "MaxLength": "18",
    "Default": "0.0.0.0/0",
    "AllowedPattern": "(\\d{1,3})\\. (\\d{1,3})\\. (\\d{1,3})\\. (\\d{1,3})/(\\d{1,2})",
    "ConstraintDescription": "must be a valid IP CIDR range of the form x.x.x.x/x."
  },
  "UseSSL": {
    "AllowedValues": ["Yes", "No"],
    "Default": "No",
    "Description": "Select \"Yes\" to implement SSL, \"No\" to skip (default).",
    "Type": "String"
  },
  "ALBSSLCertificateARN": {
    "Default": "",
    "Description": "[Optional] The ARN of the SSL certificate to be used for the
Application Load Balancer",
    "Type": "String"
  },
  "HostedZoneName": {
    "AllowedPattern": "^$|(([a-zA-Z0-9]|[a-zA-Z0-9][a-zA-Z0-9\\-]*[a-zA-Z0-9])\\
\\.)*([A-Za-z0-9]|[A-Za-z0-9][A-Za-z0-9\\-]*[A-Za-z0-9])$",
    "Default": "",
    "Description": "[Optional] The domain name of a valid Hosted Zone on AWS.",
    "Type": "String"
  }
},
"Conditions": {
  "UseALBSSL": {"Fn::Equals": [{"Ref": "UseSSL"}, "Yes"]}
},

```

```

"Rules": {
  "SubnetsInVPC": {
    "Assertions": [
      {
        "Assert": {"Fn::EachMemberEquals": [{"Fn::ValueOf": ["Subnets", "VpcId"]}, {"Ref": "VpcId"}]},
        "AssertDescription": "All subnets must be in the VPC"
      }
    ]
  },
  "ValidateHostedZone": {
    "RuleCondition": {"Fn::Equals": [{"Ref": "UseSSL"}, "Yes"]},
    "Assertions": [
      {
        "Assert": {"Fn::Not": [{"Fn::Equals": [{"Ref": "ALBSSLCertificateARN"}, ""]}]},
        "AssertDescription": "ACM Certificate value cannot be empty if SSL is required"
      },
      {
        "Assert": {"Fn::Not": [{"Fn::Equals": [{"Ref": "HostedZoneName"}, ""]}]},
        "AssertDescription": "Route53 Hosted Zone Name is mandatory when SSL is required"
      }
    ]
  }
},
"Resources": {
  "WebServerGroup": {
    "Type": "AWS::AutoScaling::AutoScalingGroup",
    "Properties": {
      "VPCZoneIdentifier": {"Ref": "Subnets"},
      "LaunchTemplate": {
        "LaunchTemplateId": {"Ref": "LaunchTemplate"},
        "Version": {"Fn::GetAtt": ["LaunchTemplate", "LatestVersionNumber"]}
      },
      "MinSize": "2",
      "MaxSize": "2",
      "TargetGroupARNs": [{"Ref": "ALBTargetGroup"}]
    },
    "CreationPolicy": {
      "ResourceSignal": {"Timeout": "PT15M"}
    },
    "UpdatePolicy": {

```

```

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

```

```

        "mode": "000400",
        "owner": "root",
        "group": "root"
    }
},
"services": {
    "sysvinit": {
        "httpd": {
            "enabled": "true",
            "ensureRunning": "true"
        },
        "cfn-hup": {
            "enabled": "true",
            "ensureRunning": "true",
            "files": [
                "/etc/cfn/cfn-hup.conf",
                "/etc/cfn/hooks.d/cfn-auto-reloader.conf"
            ]
        }
    }
}
},
"Properties": {
    "LaunchTemplateName": {"Fn::Sub": "${AWS::StackName}-launch-template"},
    "LaunchTemplateData": {
        "ImageId": "{{resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2}}",
        "SecurityGroupIds": [{"Ref": "InstanceSecurityGroup"}],
        "InstanceType": {"Ref": "InstanceType"},
        "KeyName": {"Ref": "KeyName"},
        "UserData": {
            "Fn::Base64": {"Fn::Join": ["", [
                "#!/bin/bash\n",
                "yum install -y aws-cfn-bootstrap\n",
                "/opt/aws/bin/cfn-init -v ",
                "    --stack ", {"Ref": "AWS::StackName"},
                "    --resource LaunchTemplate ",
                "    --region ", {"Ref": "AWS::Region"}, "\n",
                "/opt/aws/bin/cfn-signal -e $? ",
                "    --stack ", {"Ref": "AWS::StackName"},
                "    --resource WebServerGroup ",
                "    --region ", {"Ref": "AWS::Region"}, "\n"
            ]]}
        }
    }
}

```

```

        ]}]
    }
}
},
"ELBSecurityGroup": {
    "Type": "AWS::EC2::SecurityGroup",
    "Properties": {
        "GroupDescription": "Allow access to the ELB",
        "VpcId": {"Ref": "VpcId"},
        "SecurityGroupIngress": [{
            "Fn::If": [
                "UseALBSSL",
                {
                    "IpProtocol": "tcp",
                    "FromPort": 443,
                    "ToPort": 443,
                    "CidrIp": "0.0.0.0/0"
                },
                {
                    "IpProtocol": "tcp",
                    "FromPort": 80,
                    "ToPort": 80,
                    "CidrIp": "0.0.0.0/0"
                }
            ]
        }]
    }
},
"ApplicationLoadBalancer": {
    "Type": "AWS::ElasticLoadBalancingV2::LoadBalancer",
    "Properties": {
        "Subnets": {"Ref": "Subnets"},
        "SecurityGroups": [{"Ref": "ELBSecurityGroup"}]
    }
},
"ALBListener": {
    "Type": "AWS::ElasticLoadBalancingV2::Listener",
    "Properties": {
        "DefaultActions": [{
            "Type": "forward",
            "TargetGroupArn": {"Ref": "ALBTargetGroup"}
        }],
        "LoadBalancerArn": {"Ref": "ApplicationLoadBalancer"},

```

```

    "Port": {"Fn::If": ["UseALBSSL", 443, 80]},
    "Protocol": {"Fn::If": ["UseALBSSL", "HTTPS", "HTTP"]},
    "Certificates": [{
      "Fn::If": [
        "UseALBSSL",
        {"CertificateArn": {"Ref": "ALBSSLCertificateARN"}},
        {"Ref": "AWS::NoValue"}
      ]
    }]
  },
  "ALBTargetGroup": {
    "Type": "AWS::ElasticLoadBalancingV2::TargetGroup",
    "Properties": {
      "HealthCheckIntervalSeconds": 30,
      "HealthCheckTimeoutSeconds": 5,
      "HealthyThresholdCount": 3,
      "Port": 80,
      "Protocol": "HTTP",
      "UnhealthyThresholdCount": 5,
      "VpcId": {"Ref": "VpcId"}
    }
  },
  "InstanceSecurityGroup": {
    "Type": "AWS::EC2::SecurityGroup",
    "Properties": {
      "GroupDescription": "Enable SSH access and HTTP access on the inbound port",
      "SecurityGroupIngress": [
        {
          "IpProtocol": "tcp",
          "FromPort": 80,
          "ToPort": 80,
          "SourceSecurityGroupId": {"Fn::Select": [0, {"Fn::GetAtt":
["ApplicationLoadBalancer", "SecurityGroups"]}]}
        },
        {
          "IpProtocol": "tcp",
          "FromPort": 22,
          "ToPort": 22,
          "CidrIp": {"Ref": "SSHLocation"}
        }
      ],
      "VpcId": {"Ref": "VpcId"}
    }
  }
}

```

```

    },
    "RecordSet": {
      "Type": "AWS::Route53::RecordSetGroup",
      "Condition": "UseALBSSL",
      "Properties": {
        "HostedZoneName": {"Fn::Join": ["", [{"Ref": "HostedZoneName"}, "."]]},
        "RecordSets": [{
          "Name": {"Fn::Join": ["", [
            {"Fn::Select": ["0", {"Fn::Split": [".", {"Fn::GetAtt":
["ApplicationLoadBalancer", "DNSName"]}]}}]},
            ".",
            {"Ref": "HostedZoneName"},
            "."
          ]]},
          "Type": "A",
          "AliasTarget": {
            "DNSName": {"Fn::GetAtt": ["ApplicationLoadBalancer", "DNSName"]},
            "EvaluateTargetHealth": true,
            "HostedZoneId": {"Fn::GetAtt": ["ApplicationLoadBalancer",
"CanonicalHostedZoneID"]}
          }
        ]}
      }
    },
    "Outputs": {
      "URL": {
        "Description": "URL of the website",
        "Value": {"Fn::Join": ["", [
          {"Fn::If": [
            "UseALBSSL",
            {"Fn::Join": ["", [
              "https://",
              {"Fn::Join": ["", [
                {"Fn::Select": ["0", {"Fn::Split": [".", {"Fn::GetAtt":
["ApplicationLoadBalancer", "DNSName"]}]}}]},
                ".",
                {"Ref": "HostedZoneName"},
                "."
              ]]}
            ]]},
            {"Fn::Join": ["", [
              "http://",
              {"Fn::GetAtt": ["ApplicationLoadBalancer", "DNSName"]}
            ]]}
          ]]}
        ]}
      }
    }
  },
  "Outputs": {
    "URL": {
      "Description": "URL of the website",
      "Value": {"Fn::Join": ["", [
        {"Fn::If": [
          "UseALBSSL",
          {"Fn::Join": ["", [
            "https://",
            {"Fn::Join": ["", [
              {"Fn::Select": ["0", {"Fn::Split": [".", {"Fn::GetAtt":
["ApplicationLoadBalancer", "DNSName"]}]}}]},
              ".",
              {"Ref": "HostedZoneName"},
              "."
            ]]}
          ]]},
          {"Fn::Join": ["", [
            "http://",
            {"Fn::GetAtt": ["ApplicationLoadBalancer", "DNSName"]}
          ]]}
        ]]}
      ]}
    }
  }
}

```



```

    ]}]
  ]}
  ]}]
}
}
}

```

YAML

```

AWSTemplateFormatVersion: 2010-09-09
Parameters:
  VpcId:
    Type: AWS::EC2::VPC::Id
    Description: VpcId of your existing Virtual Private Cloud (VPC)
    ConstraintDescription: must be the VPC Id of an existing Virtual Private Cloud.
  Subnets:
    Type: List<AWS::EC2::Subnet::Id>
    Description: The list of SubnetIds in your Virtual Private Cloud (VPC)
    ConstraintDescription: >-
      must be a list of at least two existing subnets associated with at least
      two different availability zones. They should be residing in the selected
      Virtual Private Cloud.
  InstanceType:
    Description: WebServer EC2 instance type
    Type: String
    Default: t2.micro
    AllowedValues:
      - t2.micro
      - t3.micro
    ConstraintDescription: must be a valid EC2 instance type.
  KeyName:
    Description: Name of an existing EC2 KeyPair to enable SSH access to the instances
    Type: AWS::EC2::KeyPair::KeyName
    ConstraintDescription: must be the name of an existing EC2 KeyPair.
  SSHLocation:
    Description: The IP address range that can be used to SSH to the EC2 instances
    Type: String
    MinLength: '9'
    MaxLength: '18'
    Default: 0.0.0.0/0
    AllowedPattern: '(\d{1,3})\.\d{1,3})\.\d{1,3})\.\d{1,3})/(\d{1,2})'
    ConstraintDescription: must be a valid IP CIDR range of the form x.x.x.x/x.
  UseSSL:

```

```

AllowedValues:
  - 'Yes'
  - 'No'
ConstraintDescription: Select Yes to create a HTTPS Listener
Default: 'No'
Description: 'Select "Yes" to implement SSL, "No" to skip (default).'
```

Type: String

```

ALBSSLCertificateARN:
  Default: ''
  Description: >-
    [Optional] The ARN of the SSL certificate to be used for the Application
    Load Balancer
  Type: String
HostedZoneName:
  AllowedPattern: >-
    ^$|((([a-zA-Z0-9]|[a-zA-Z0-9][a-zA-Z0-9\-\-]*[a-zA-Z0-9])\.)+)([A-Za-z0-9]|[A-Za-
z0-9][A-Za-z0-9\-\-]*[A-Za-z0-9]))$
  Default: ''
  Description: '[Optional] The domain name of a valid Hosted Zone on AWS.'
```

Type: String

```

Conditions:
  UseALBSSL: !Equals
    - !Ref UseSSL
    - 'Yes'
Rules:
  SubnetsInVPC:
    Assertions:
      - Assert:
          'Fn::EachMemberEquals':
            - 'Fn::ValueOf':
                - Subnets
                - VpcId
            - Ref: VpcId
          AssertDescription: All subnets must be in the VPC
  ValidateHostedZone:
    RuleCondition: !Equals
      - !Ref UseSSL
      - 'Yes'
    Assertions:
      - Assert: !Not
          - !Equals
            - !Ref ALBSSLCertificateARN
            - ''
          AssertDescription: ACM Certificate value cannot be empty if SSL is required
```

- Assert: !Not
 - !Equals
 - !Ref HostedZoneName
 - ''

AssertDescription: Route53 Hosted Zone Name is mandatory when SSL is required

Resources:

WebServerGroup:

Type: AWS::AutoScaling::AutoScalingGroup

Properties:

VPCZoneIdentifier: !Ref Subnets

LaunchTemplate:

LaunchTemplateId: !Ref LaunchTemplate

Version: !GetAtt LaunchTemplate.LatestVersionNumber

MinSize: '2'

MaxSize: '2'

TargetGroupARNs:

- !Ref ALBTargetGroup

CreationPolicy:

ResourceSignal:

Timeout: PT15M

UpdatePolicy:

AutoScalingRollingUpdate:

MinInstancesInService: '1'

MaxBatchSize: '1'

PauseTime: PT15M

WaitOnResourceSignals: 'true'

LaunchTemplate:

Type: AWS::EC2::LaunchTemplate

Metadata:

Comment: Install a simple application

AWS::CloudFormation::Init:

config:

packages:

yum:

httpd: []

files:

/var/www/html/index.html:

content: !Join

- |+

- - >-

<h1>Congratulations, you have successfully launched the AWS
CloudFormation sample.</h1>

mode: '000644'

owner: root

```

    group: root
    /etc/cfn/cfn-hup.conf:
      content: !Sub |
        [main]
        stack=${AWS::StackId}
        region=${AWS::Region}
      mode: '000400'
      owner: root
      group: root
    /etc/cfn/hooks.d/cfn-auto-reloader.conf:
      content: !Sub |-
        [cfn-auto-reloader-hook]
        triggers=post.update
        path=Resources.LaunchTemplate.Metadata.AWS::CloudFormation::Init
        action=/opt/aws/bin/cfn-init -v --stack ${AWS::StackName} --resource
LaunchTemplate --region ${AWS::Region}
        runas=root
      mode: '000400'
      owner: root
      group: root
    services:
      sysvinit:
        httpd:
          enabled: 'true'
          ensureRunning: 'true'
        cfn-hup:
          enabled: 'true'
          ensureRunning: 'true'
        files:
          - /etc/cfn/cfn-hup.conf
          - /etc/cfn/hooks.d/cfn-auto-reloader.conf
  Properties:
    LaunchTemplateName: !Sub ${AWS::StackName}-launch-template
    LaunchTemplateData:
      ImageId: '{{resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-
x86_64-gp2}}'
      SecurityGroupIds:
        - !Ref InstanceSecurityGroup
      InstanceType: !Ref InstanceType
      KeyName: !Ref KeyName
      UserData: !Base64
        Fn::Sub: |
          #!/bin/bash
          yum install -y aws-cfn-bootstrap

```

```
    /opt/aws/bin/cfn-init -v --stack ${AWS::StackName} --resource
LaunchTemplate --region ${AWS::Region}
    /opt/aws/bin/cfn-signal -e $? --stack ${AWS::StackName} --resource
WebServerGroup --region ${AWS::Region}
ELBSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Allow access to the ELB
    VpcId: !Ref VpcId
    SecurityGroupIngress:
      - !If
        - UseALBSSL
        - IpProtocol: tcp
          FromPort: 443
          ToPort: 443
          CidrIp: 0.0.0.0/0
        - IpProtocol: tcp
          FromPort: 80
          ToPort: 80
          CidrIp: 0.0.0.0/0
ApplicationLoadBalancer:
  Type: AWS::ElasticLoadBalancingV2::LoadBalancer
  Properties:
    Subnets: !Ref Subnets
    SecurityGroups:
      - !Ref ELBSecurityGroup
ALBListener:
  Type: AWS::ElasticLoadBalancingV2::Listener
  Properties:
    DefaultActions:
      - Type: forward
        TargetGroupArn: !Ref ALBTargetGroup
    LoadBalancerArn: !Ref ApplicationLoadBalancer
    Port: !If
      - UseALBSSL
      - 443
      - 80
    Protocol: !If
      - UseALBSSL
      - HTTPS
      - HTTP
    Certificates:
      - !If
        - UseALBSSL
```

```

    - CertificateArn: !Ref ALBSSLCertificateARN
    - !Ref 'AWS::NoValue'
ALBTargetGroup:
  Type: AWS::ElasticLoadBalancingV2::TargetGroup
  Properties:
    HealthCheckIntervalSeconds: 30
    HealthCheckTimeoutSeconds: 5
    HealthyThresholdCount: 3
    Port: 80
    Protocol: HTTP
    UnhealthyThresholdCount: 5
    VpcId: !Ref VpcId
InstanceSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Enable SSH access and HTTP access on the inbound port
    SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: 80
        ToPort: 80
        SourceSecurityGroupId: !Select
          - 0
          - !GetAtt
            - ApplicationLoadBalancer
            - SecurityGroups
      - IpProtocol: tcp
        FromPort: 22
        ToPort: 22
        CidrIp: !Ref SSHLocation
    VpcId: !Ref VpcId
RecordSet:
  Type: AWS::Route53::RecordSetGroup
  Condition: UseALBSSL
  Properties:
    HostedZoneName: !Join
      - ''
      - - !Ref HostedZoneName
        - .
    RecordSets:
      - Name: !Join
          - ''
          - - !Select
              - '0'
              - !Split

```

```

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

```

- DNSName

Sintaxis de Conditions de la plantilla de CloudFormation

La sección opcional `Conditions` contiene declaraciones que definen las circunstancias por las que se crean o configuran las entidades. Por ejemplo, puede crear una condición y asociarla a un recurso o salida de manera que CloudFormation cree el recurso o salida solo si la condición es verdadera. Del mismo modo, puede asociar una condición a una propiedad de manera que CloudFormation establezca la propiedad en un valor específico solo si la condición es verdadera. Si la condición es falsa, CloudFormation establece la propiedad en un valor alternativo que especifica.

Puede utilizar condiciones cuando desee reutilizar una plantilla para crear recursos en diferentes contextos, como, por ejemplo, entornos de pruebas frente a entornos de producción. Por ejemplo, en su plantilla, puede agregar un parámetro de entrada `EnvironmentType` que acepte `prod` o `test` como entradas. Para el entorno `prod`, podría incluir instancias de EC2 con determinadas capacidades; sin embargo, para el entorno `test`, podría utilizar capacidades reducidas para ahorrar dinero. La definición de esta condición le permite definir qué recursos se crean y cómo se configuran para cada tipo de entorno.

Sintaxis

La sección `Conditions` consta del nombre de clave `Conditions`. Cada declaración de condición incluye un ID lógico y una o más funciones intrínsecas.

JSON

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

YAML

```
Conditions:  
  LogicalConditionName1:
```



```
Intrinsic function:
```

```
...
```

```
LogicalConditionName2:
```

```
Intrinsic function:
```

```
...
```

Cómo funcionan las condiciones

Para utilizar condiciones, siga estos pasos:

1. Agregar una definición de parámetro: defina las entradas que evaluarán sus condiciones en la sección `Parameters` de su plantilla. Las condiciones se evalúan en `true` o `false` en función de los valores de estos parámetros de entrada. Tenga en cuenta que los seudoparámetros están disponibles automáticamente y no requieren una definición explícita en la sección `Parameters`. Para obtener más información sobre pseudoparámetros, consulte . [Obtención de valores de AWS mediante pseudoparámetros](#).

2. Agregar una definición de condición: defina las condiciones en la sección `Conditions` mediante funciones intrínsecas como `Fn::If` o `Fn::Equals`. Las condiciones determinan cuándo CloudFormation crea los recursos asociados. Las condiciones se pueden basar en lo siguiente:

- Valores de entrada o de seudoparámetros
- Otras condiciones
- Valores de asignación

Sin embargo, no puede hacer referencia a los ID lógicos de los recursos ni a sus atributos en las condiciones.

3. Asociar las condiciones a recursos o salidas: haga referencia a las condiciones en los recursos o las salidas mediante la clave `Condition` y el ID lógico de una condición. También puede usar `Fn::If` en otras partes de la plantilla (como los valores de las propiedades) para establecer valores en función de una condición. Para obtener más información, consulte [Utilización de la clave `Condition`](#).

CloudFormation evalúa las condiciones al crear o actualizar una pila. CloudFormation crea entidades que se asocian a una condición `true` e ignora las entidades que están asociadas a una condición `false`. Además, CloudFormation reevalúa estas condiciones durante las actualizaciones de una pila antes de modificar los recursos. Las entidades que permanecen asociadas a una condición verdadera se actualizan, mientras que las que se asocian a una condición falsa se eliminan.

 Important

Durante la actualización de una pila, no se pueden actualizar condiciones por sí mismas. Puede actualizar condiciones solo cuando incluye cambios que añaden, modifican o eliminan recursos.

Funciones intrínsecas de condiciones

Puede utilizar las siguientes funciones intrínsecas para definir condiciones:

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

 Note

`Fn::If` solo se admite en el atributo de metadatos, el atributo de la política de actualización y los valores de la propiedad en la sección `Resources` y en la sección `Outputs` de una plantilla.

Utilización de la clave **Condition**

Una vez que se define una condición, puede aplicarla en varios lugares de la plantilla, como `Resources` y `Outputs`, mediante la clave `Condition`. La clave `Condition` hace referencia al nombre lógico de una condición y devuelve el resultado evaluado de la condición especificada.

Temas

- [Asociación de condiciones a los recursos](#)
- [Asociación de condiciones a las salidas](#)
- [Condiciones de referencia en otras condiciones](#)

- [Devuelva condicionalmente los valores de las propiedades mediante Fn::If](#)

Asociación de condiciones a los recursos

Para crear recursos de forma condicional, añada la clave `Condition` y el ID lógico de la condición como un atributo del recurso. CloudFormation crea el recurso únicamente cuando la condición se evalúa como `true`.

JSON

```
"NewVolume" : {
  "Type" : "AWS::EC2::Volume",
  "Condition" : "IsProduction",
  "Properties" : {
    "Size" : "100",
    "AvailabilityZone" : { "Fn::GetAtt" : [ "EC2Instance", "AvailabilityZone" ] }
  }
}
```

YAML

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

Asociación de condiciones a las salidas

También puede asociar condiciones a las salidas. CloudFormation crea la salida únicamente cuando la condición asociada se evalúa como verdadera.

JSON

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

```
}
```

YAML

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

Condiciones de referencia en otras condiciones

Al definir las condiciones en la sección `Conditions`, puede hacer referencia a otras condiciones mediante la clave `Condition`. Esto le permite crear una lógica condicional más compleja mediante la combinación de varias condiciones.

En el siguiente ejemplo, la condición `IsProdAndFeatureEnabled` se evalúa como verdadera solo si las condiciones `IsProduction` y `IsFeatureEnabled` se evalúan como verdaderas.

JSON

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

YAML

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

Devuelva condicionalmente los valores de las propiedades mediante **Fn::If**

Para un control más granular, puede utilizar la función intrínseca `Fn::If` para devolver condicionalmente uno de los dos valores de propiedad de los recursos o las salidas. Esta función evalúa una condición y devuelve un valor si la condición es verdadera y otro valor si la condición es falsa.

Valores condicionales de la propiedad

En el siguiente ejemplo, se muestra la configuración de un tipo de instancia de EC2 en función de una condición del entorno. Si la condición `IsProduction` es verdadera, el tipo de instancia se establece en `c5.xlarge`. De lo contrario, se establece en `t3.small`.

JSON

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

YAML

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

Eliminación condicional de propiedades

Puede utilizar el pseudoparámetro `AWS::NoValue` como valor devuelto para eliminar la propiedad correspondiente cuando una condición es falsa.

JSON

```
"DBSnapshotIdentifier" : {
  "Fn::If" : [
```

```
    "UseDBSnapshot",
    {"Ref" : "DBSnapshotName"},
    {"Ref" : "AWS::NoValue"}
  ]
}
```

YAML

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

Ejemplos

Temas

- [Creación de recursos en función del entorno](#)
- [Aprovisionamiento de recursos con varias condiciones](#)

Creación de recursos en función del entorno

En los siguientes ejemplos se aprovisiona una instancia de EC2 y se crea y adjunta condicionalmente un nuevo volumen de EBS solo si el tipo de entorno es prod. Si el entorno es test, simplemente se crea la instancia de EC2 sin el volumen adicional.

JSON

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

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

```
}  
}
```

YAML

```
AWSTemplateFormatVersion: 2010-09-09  
Parameters:  
  EnvType:  
    Description: Environment type  
    Default: test  
    Type: String  
    AllowedValues:  
      - prod  
      - test  
    ConstraintDescription: must specify prod or test  
Conditions:  
  IsProduction: !Equals  
    - !Ref EnvType  
    - prod  
Resources:  
  EC2Instance:  
    Type: AWS::EC2::Instance  
    Properties:  
      ImageId: ami-1234567890abcdef0  
      InstanceType: c5.xlarge  
  MountPoint:  
    Type: AWS::EC2::VolumeAttachment  
    Condition: IsProduction  
    Properties:  
      InstanceId: !Ref EC2Instance  
      VolumeId: !Ref NewVolume  
      Device: /dev/sdh  
  NewVolume:  
    Type: AWS::EC2::Volume  
    Condition: IsProduction  
    Properties:  
      Size: 100  
      AvailabilityZone: !GetAtt  
        - EC2Instance  
        - AvailabilityZone
```


Aprovisionamiento de recursos con varias condiciones

En los siguientes ejemplos se crea condicionalmente un bucket de S3 si se proporciona un nombre de bucket y se adjunta una política de bucket solo cuando el entorno está definido como prod. Si no se proporciona ningún nombre de bucket o el entorno es test, no se crea ningún recurso.

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: ...
```

En este ejemplo, la condición `CreateBucketPolicy` demuestra cómo hacer referencia a otras condiciones mediante la clave `Condition`. La política se crea solo cuando tanto las condiciones `IsProduction` y `CreateBucket` se evalúan como verdaderas.

Note

Para ver ejemplos más complejos del uso de condiciones, consulte el tema de [atributos de Condition](#) en la Guía de referencia de plantillas de CloudFormation.

Sección Transform de la plantilla de CloudFormation

La sección `Transform` opcional especifica una o varias macros que CloudFormation utiliza para procesar su plantilla de alguna manera.

Las macros pueden llevar a cabo tareas sencillas, como buscar y reemplazar texto o pueden efectuar transformaciones más amplias en toda la plantilla. CloudFormation ejecuta macros en el orden en que se especifican. Cuando se crea un conjunto de cambios, CloudFormation genera un conjunto de cambios que incluyen el contenido de la plantilla procesada. A continuación, puede revisar los cambios y ejecutar el conjunto de cambios. Para obtener más información sobre cómo funcionan las macros, consulte [Procesamientos personalizados en plantillas de CloudFormation con macros de plantillas](#).

CloudFormation también admite transformaciones, que son macros alojadas en CloudFormation. CloudFormation trata estas transformaciones de la misma manera que cualquier macro que cree en términos de orden de ejecución y ámbito. Para obtener más información, consulte [Referencia de transformación](#).

Para declarar varias macros, utilice un formato de lista y especifique una o varias macros.

Por ejemplo, en el ejemplo de plantilla siguiente, CloudFormation evalúa MyMacro y, a continuación, AWS::Serverless, los cuales pueden procesar el contenido de toda la plantilla porque están incluidos en la sección Transform.

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

Sintaxis de la versión del formato de plantilla de CloudFormation

La sección `AWSTemplateFormatVersion` (opcional) identifica la versión del formato de la plantilla a la que se ajusta la plantilla. La última versión de formato de plantilla es 2010-09-09 y es en la actualidad el único valor válido.

La versión del formato de plantilla no es la misma que la versión de la API. La versión del formato de la plantilla puede cambiar de manera independiente a las versiones de la API.

El valor para la declaración de versión de formato de plantilla debe ser una cadena literal. No puede utilizar un parámetro o una función para especificar la versión de formato de plantilla. Si no especifica un valor, CloudFormation asume la última versión de formato de plantilla. El siguiente fragmento de código es un ejemplo de una declaración válida de versión de formato de plantilla:

JSON

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

YAML

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

Sintaxis de Description de la plantilla de CloudFormation

La sección `Description` (opcional) permite incluir una cadena de texto que describe la plantilla. Esta sección debe seguir siempre la sección de la versión de formato de plantilla.

El valor de la declaración de la descripción debe ser una cadena literal de entre 0 y 1024 bytes de longitud. No puede utilizar un parámetro o función para especificar la descripción. El siguiente fragmento de código es un ejemplo de una declaración de descripción:

Important

Durante la actualización de una pila, no puede actualizar la sección `Description` por sí misma. Puede actualizarla solo cuando incluye cambios que añaden, modifican o eliminan recursos.

JSON

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

YAML

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

Creación de plantillas visualmente con Infrastructure Composer

AWS Infrastructure Composer (anteriormente conocido como Application Composer) lo ayuda a componer y configurar visualmente aplicaciones modernas en AWS. En lugar de escribir código, puede arrastrar y soltar diferentes recursos para crear su aplicación de forma visual.

Infrastructure Composer en el modo de consola de CloudFormation es la herramienta recomendada para trabajar visualmente con plantillas de CloudFormation. Esta versión de Infrastructure Composer, a la que puede acceder desde la consola de CloudFormation, es una mejora con respecto a una herramienta anterior denominada CloudFormation Designer.

Con Infrastructure Composer en el modo de consola de CloudFormation, puede arrastrar, soltar, configurar y conectar una variedad de recursos, denominados tarjetas, en un lienzo visual. Este enfoque visual facilita el diseño y la edición de la arquitectura de la aplicación sin tener que trabajar directamente con las plantillas. Para acceder a este modo desde la [consola de CloudFormation](#), seleccione Infrastructure Composer en el menú de navegación de la izquierda.

Para obtener más información, consulte [Cómo componer en AWS Infrastructure Composer](#) en la Guía del usuario de AWS Infrastructure Composer.

¿Por qué utilizar Infrastructure Composer en el modo de consola de CloudFormation?

Visualizar plantillas en Infrastructure Composer lo ayuda a identificar brechas y áreas de mejora en las plantillas y en la arquitectura de las aplicaciones de CloudFormation. Infrastructure Composer mejora la experiencia de desarrollo con la facilidad y la eficiencia de crear y modificar visualmente las pilas de CloudFormation. Puede empezar con un borrador inicial, crear un código implementable e incorporar los flujos de trabajo de desarrollador con el diseñador visual de Infrastructure Composer.

¿En qué se diferencia este modo de la consola de Infrastructure Composer?

Si bien la versión de consola de CloudFormation de Infrastructure Composer tiene características similares a las de la consola estándar de Infrastructure Composer, existen algunas diferencias. Las tarjetas relacionadas con Lambda (la función de Lambda y la capa de Lambda) requieren conjuntos de códigos y soluciones de empaquetado que no están disponibles en Infrastructure Composer en el modo de consola de CloudFormation. La sincronización local tampoco está disponible en este modo.

Sin embargo, puede utilizar estas tarjetas relacionadas con Lambda y la característica de sincronización local en la [consola de Infrastructure Composer](#) o en el AWS Toolkit for Visual Studio Code. Para más información, consulte [Guía de desarrolladores de AWS Infrastructure Composer](#) e [Infrastructure Composer](#) en la Guía del usuario de AWS Toolkit for Visual Studio Code.

Servidor de lenguaje AWS CloudFormation

El servidor de lenguaje AWS CloudFormation ofrece capacidades para acelerar la creación de infraestructura como código (IaC) y la implementación de recursos de AWS de forma segura y confiable. Sigue el [protocolo de servidor de lenguaje](#) (LSP) para proporcionar documentación sobre cómo pasar el ratón, completar automáticamente, realizar diagnósticos mediante validación estática, ir a la definición y codificar acciones. Además de estas capacidades de servidor de lenguaje tradicional, el servidor añade características en línea para explorar e implementar recursos de AWS a través de CloudFormation. Esto incluye la capacidad de validar e implementar plantillas mediante conjuntos de cambios; ver las diferencias, los eventos, los recursos y las salidas de las pilas; enumerar las pilas y buscar los recursos por tipo, e insertar el estado de los recursos en tiempo real directamente en las plantillas de CloudFormation. Las herramientas para desarrolladores y los entornos de desarrollo integrado (IDE) pueden aprovechar las capacidades del servidor de lenguaje para mejorar la experiencia de los desarrolladores de IaC en CloudFormation.

Temas

- [Los IDE se integran con el servidor de lenguaje AWS CloudFormation](#)
- [Introducción](#)
- [Inicialización de un proyecto de CloudFormation en el IDE](#)
- [Código abierto](#)
- [¿Necesita ayuda?](#)

Los IDE se integran con el servidor de lenguaje AWS CloudFormation

La extensión del [Kit de herramientas de AWS](#) integra el servidor de lenguaje de CloudFormation para mejorar la experiencia de creación. Actualmente, esta integración es compatible con los IDE basados en VS Code, que incluyen:

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

Introducción

Temas

- [Requisitos previos](#)
- [Paso 1: instalar o actualizar el Kit de herramientas de AWS](#)
- [Paso 2: acceder a CloudFormation en el panel del kit de herramientas de AWS](#)
- [Paso 3: validar, probar y refinar la plantilla](#)
- [Paso 4: navegar por la plantilla](#)
- [Paso 5: validar e implementar](#)

Requisitos previos

Antes de comenzar, asegúrese de que:

- Está utilizando un IDE basado en VS Code en un sistema operativo compatible (macOS, Windows o Linux).
- Ha instalado o actualizado a la última versión del [Kit de herramientas de AWS](#) para su IDE.

Algunas características del servidor de lenguaje AWS CloudFormation requieren una Cuenta de AWS activa y credenciales configuradas. Debe iniciar sesión en su Cuenta de AWS a través del Kit de herramientas de AWS con credenciales válidas.

Paso 1: instalar o actualizar el Kit de herramientas de AWS

- Abra el administrador de extensiones o complementos de su IDE.

- Busque el Kit de herramientas de AWS.
- Instale o actualice a la versión 3.85.0 o posterior del Kit de herramientas de AWS de su IDE.
- Reinicie el IDE tras la instalación.

Nota: Tras la instalación, el Kit de herramientas de AWS habilita automáticamente la compatibilidad con el IDE de CloudFormation. La primera vez que instale o actualice el Kit de herramientas de AWS con la extensión de servidor de lenguaje AWS CloudFormation, se le solicitará que conceda permiso a AWS para recopilar datos de uso anónimos. Estos datos ayudan a que AWS mejore el servidor de lenguaje de CloudFormation y mejoran la experiencia de creación. No se recopila información confidencial y AWS no registra ni almacena el contenido de las plantillas, las configuraciones de los recursos ni ningún dato identificable del cliente. Puede cambiar las preferencias de telemetría en cualquier momento desde la configuración del IDE. Los datos de uso recopilados se centran únicamente en las interacciones de las características y las métricas de rendimiento. Esta información ayuda a AWS a identificar y priorizar mejoras, como una validación más rápida, un mejor autocompletado y un mejor diagnóstico de errores.

Paso 2: acceder a CloudFormation en el panel del kit de herramientas de AWS

En su IDE, abra el panel del Kit de herramientas de AWS desde la barra de actividades. En el Kit de herramientas de AWS, seleccione CLOUDFORMATION.

Secciones del panel:

- Región: muestra la Región de AWS actual. Para cambiarla, seleccione el nombre de la región o utilice el comando AWS CloudFormation: seleccionar región de la paleta de comandos.
- Entorno: indica el entorno que seleccionó durante CFN init (por ejemplo, desarrollo, pruebas o producción). Este valor aparece solo después de completar el paso de selección del entorno.
- Pilas: muestra una lista paginada de las pilas de CloudFormation de su cuenta.
 - Haga clic en el icono de actualización para actualizar la lista de pilas.
 - Use el icono + para implementar una nueva plantilla como pila.
 - Cada entrada de la pila incluye:
 - Descripción general: muestra el resumen y el estado de la pila
 - ID de pila de
 - Descripción
 - Hora de creación y Hora de actualización

- Estado y Motivo del estado
- Eventos, Salidas y Recursos
- Conjunto de cambios
- Recursos: después de seleccionar un tipo de recurso, el panel muestra los recursos de AWS de ese tipo en su cuenta. Puede verlos, actualizarlos, copiarlos o importarlos a su plantilla.

Paso 3: validar, probar y refinar la plantilla

A medida que redacta la plantilla de CloudFormation, el IDE proporciona asistencia de creación inteligente para ayudarle a crear una infraestructura precisa y compatible con mayor rapidez. El servidor de lenguaje de CloudFormation se ejecuta en segundo plano y proporciona las siguientes características de creación:

- Finalización de código: sugiere tipos de recursos, parámetros y propiedades en función de los esquemas de CloudFormation.
- Añadir recursos de AWS existentes: le permite importar los recursos existentes de su Cuenta de AWS a su plantilla. El IDE usa el [API de control de nube de AWS \(CCAPI\)](#) para recuperar la configuración activa y las propiedades del recurso, lo que le ayuda a clonar o reutilizar la infraestructura existente en su plantilla.

Adición de recursos a una plantilla

- Expanda el panel de recursos de CloudFormation: en el panel lateral del Kit de herramientas de AWS, expanda CloudFormation y, en Recursos, haga clic en el icono Añadir +.
- Busque un tipo de recurso: para buscar un tipo de recurso de AWS específico, en la barra de búsqueda escriba el tipo de recurso de AWS específico que desee añadir. Ejemplo:
 - `AWS::S3::Bucket`
 - `AWS::Lambda::Function`

Puede seleccionar el tipo de recurso mediante AWS CloudFormation: añadir tipos de recursos en la paleta de comandos.

- Añadir recursos a la plantilla: en el panel Recursos, se muestra una lista paginada de los recursos de AWS detectados en su cuenta. Si tiene muchos recursos, solo se muestra la primera página. Use los controles de navegación de la parte inferior del panel para desplazarse por páginas adicionales y ver todos los recursos disponibles.

- Elija el recurso que desea incluir en la plantilla.
- Puede insertar un recurso en la plantilla en dos formas, en función de su objetivo:
 - Clonar un recurso existente: para crear un recurso nuevo en su plantilla, utilice la configuración activa y las propiedades de un recurso de AWS existente.
 - Importar un recurso existente: para insertar el recurso real en su pila, añádalo a su plantilla utilizando su estado activo.

Consejos

- Puede actualizar el panel Recursos en cualquier momento para ver la lista más reciente de recursos disponibles en su cuenta o región.
- Si va a importar recursos, no añada un recurso que ya pertenezca a una pila de CloudFormation existente en la misma cuenta.
- Para confirmar si CloudFormation ya administra un recurso, haga clic en el icono i (información) situado junto al nombre del recurso.
- Como alternativa, puede usar el comando AWS CloudFormation: obtener información de administración de la pila para comprobar si un recurso pertenece a una pila.

Adición de recursos relacionados

Puede añadir recursos relacionados al recurso seleccionado mediante el comando AWS CloudFormation: añadir recursos relacionados por tipo. Una vez que haya seleccionado un tipo de recurso entre los que ya están definidos en la plantilla, el IDE muestra una lista de los recursos que normalmente están asociados a ese tipo o que dependen de él. Por ejemplo, si selecciona `AWS::EC2::Instance`, el IDE puede sugerirle que añada recursos relacionados, como `AWS::EC2::SecurityGroup` o `AWS::EC2::Subnet`. Esta característica le ayuda a crear rápidamente componentes de infraestructura conectados sin tener que buscar manualmente los tipos de recursos compatibles.

Validación estática

El IDE de CloudFormation proporciona una validación estática integrada con la tecnología de [AWS CloudFormation Linter \(cfn-lint\)](#) y [AWS CloudFormation Guard](#). Estas validaciones se llevan a cabo en segundo plano a medida que crea plantillas, lo que le ayuda a identificar los errores de sintaxis, las brechas de conformidad y los problemas relacionados con las prácticas recomendadas antes de la implementación.

Descripción general de la validación estática

Verá dos tipos de validaciones estáticas en tiempo real en el IDE:

- CloudFormation Linter (`cfn-lint`): valida la plantilla según las especificaciones de recursos y las reglas del esquema de CloudFormation.
- Guard (`cfn-guard`): valida la plantilla según las normas de cumplimiento y los paquetes de políticas organizacionales.

CloudFormation Linter (`cfn-lint`)

CloudFormation Linter está integrado en el IDE para comprobar automáticamente la sintaxis y la estructura de la plantilla a medida que escribe.

- Validación de esquemas: detecta errores de sintaxis y esquema para garantizar que las plantillas se ajusten al esquema de recursos de CloudFormation.
- Resaltado de error: muestra líneas onduladas rojas o amarillas debajo de los problemas, que representan advertencias o bloqueos de implementación.
- Pase el cursor sobre la ayuda: al pasar el cursor por encima de un error, el IDE muestra el mensaje de diagnóstico asociado a ese problema. Si hay una solución rápida disponible, también aparecerá en el panel al pasar el cursor.

Integración de CI/CD

Guard valida sus plantillas comparándolas con los conjuntos de reglas que definen las políticas de cumplimiento y seguridad. El IDE ejecuta las validaciones de Guard en tiempo real a través del servidor de lenguaje de CloudFormation, lo que le proporciona comentarios inmediatos mientras crea plantillas.

- Paquetes de reglas predeterminados: el IDE incluye un conjunto de reglas de Guard previamente registradas que se centran en las prácticas recomendadas fundamentales para la seguridad de los recursos y la higiene de la configuración. Para obtener más información, consulte [el registro de reglas de Guard](#).
- Añadir paquetes de reglas: para añadir o modificar conjuntos de reglas, abra Configuración, Guard y seleccione o carga paquetes de reglas de Guard adicionales.

Consejos: entender los indicadores de líneas onduladas.

- Garabatos azules: sugerencias de prácticas recomendadas o recomendaciones de optimización.
- Garabatos amarillos: advertencias sobre problemas no relacionados con el bloqueo (por ejemplo, falta de etiquetas o parámetros).
- Garabatos rojos: bloquean la implementación, como nombres de propiedades no válidos, campos obligatorios que faltan o discordancias en los esquemas.

Paso 4: navegar por la plantilla

Al hacer clic en un archivo de plantilla en el Explorador, el panel Esquema muestra automáticamente una vista jerárquica y estructurada de la plantilla de CloudFormation. Organiza la plantilla en secciones como `Parameters`, `Resources`, `Outputs` y `Mappings`, y muestra cada tipo de recurso e ID lógico. Esto facilita la búsqueda rápida de recursos o parámetros específicos dentro de plantillas grandes y la navegación a ellos.

Puede utilizar `Ir a la definición` para funciones intrínsecas, como `GetAtt` y `Ref`, que le permite ir directamente al recurso o parámetro al que se hace referencia en la plantilla. Esto le ayuda a rastrear las dependencias, comprender las relaciones entre los recursos y realizar modificaciones de manera más eficiente.

Paso 5: validar e implementar

Cuando esté listo para implementar la plantilla de CloudFormation, abra la paleta de comandos y ejecute `AWS CloudFormation: validar e implementar`. El comando valida la plantilla y, si no se encuentra ningún error de bloqueo, pasa a la fase de implementación. Antes de confirmarlo, el IDE muestra un resumen del [conjunto de cambios con reconocimiento de desviación](#) y una vista de diferencias para que pueda revisar todos los cambios propuestos.

Cómo funciona la validación

El IDE realiza automáticamente una [comprobación de validación antes de la implementación](#) y valida la plantilla en función de las causas de error más comunes, entre las que se incluyen las siguientes:

- Discordancias entre esquemas o sintaxis de propiedades no válidas: estos problemas suele detectarlos `cfn-lint` durante la creación, pero si un usuario continúa con la implementación sin solucionarlos, la validación de CloudFormation en el momento de la implementación mostrará los mismos errores antes de que se cree o actualice la pila.
- El nombre del recurso entra en conflicto con los recursos existentes en su cuenta.

- Restricciones específicas del servicio, como conflictos en el nombre del bucket de S3 o falta de cifrado.

Si la validación detecta errores, el IDE resalta los problemas directamente en la plantilla mediante líneas onduladas rojas o amarillas y enumera los errores en el panel PROBLEMAS. Cada problema incluye la propiedad o el recurso específico que causó el error, junto con una sugerencia de solución. Si no hay errores de bloqueo, puede pasar a la fase de implementación.

Si se encuentran advertencias (problemas no relacionados con el bloqueo), aparece un cuadro de diálogo que le permite continuar con la implementación o cancelar y realizar las correcciones necesarias.

El IDE abre un [conjunto de cambios con reconocimiento de desviación](#) en el que se muestran las diferencias entre la plantilla actual y la configuración de la pila implementada. Esto le permite revisar, confirmar o cancelar el conjunto de cambios antes de su ejecución.

Nota: Los conjuntos de cambios con reconocimiento de desviación mejoran el proceso de implementación de CloudFormation al permitirle gestionar la desviación de la pila de forma segura. La desviación de la pila se produce cuando el estado real de los recursos difiere del definido en la plantilla de CloudFormation, a menudo debido a cambios manuales realizados mediante la Consola de administración de AWS, la CLI o el SDK. El [conjunto de cambios con reconocimiento de desviación](#) de CloudFormation compara la configuración de la pila procesada con el estado actual de los recursos, y el IDE muestra estas diferencias para que pueda hacer que los recursos vuelvan a cumplir con las normas antes de la implementación.

Visualización de eventos de pilas

Cuando comience la implementación, puede ir a la pestaña CloudFormation del panel para supervisar el progreso en tiempo real. En Eventos de la pila, verá una lista de las operaciones realizadas durante la implementación. Cada evento incluye detalles como:

- Marca temporal: la hora en que ocurrió el evento.
- Recurso: el recurso de AWS específico que se crea, actualiza o elimina
- Estado: el estado actual de la operación (por ejemplo, CREATE_IN_PROGRESS, UPDATE_COMPLETE o ROLLBACK_IN_PROGRESS)
- Motivo: mensajes de contexto o de error adicionales, si corresponde

También puede ver los recursos y las salidas de la pila desde este panel. Si quiere abrir la pila en Consola de administración de AWS, utilice el icono de enlace externo situado junto al nombre de la pila. La vista Eventos de la pila le ayuda a realizar un seguimiento del progreso de la implementación, identificar posibles problemas y confirmar si la pila se ha completado correctamente.

Inicialización de un proyecto de CloudFormation en el IDE

La inicialización de un proyecto de CloudFormation en el IDE le ayuda a configurar un espacio de trabajo estructurado con las carpetas, la configuración del entorno y las credenciales de AWS correctas para que pueda validar e implementar sus plantillas de forma fiable. Puede inicializar un nuevo proyecto de CloudFormation directamente desde el IDE para crear la configuración recomendada.

Para inicializar un proyecto de CloudFormation:

- Abra la paleta de comandos
 - Desde su IDE, abra la paleta de comandos (Ctrl+Shift+P o Cmd+Shift+P en macOS).
 - Elija AWS CloudFormation: CFN Init: inicializar proyecto.
- Elija un directorio de proyectos
 - El IDE utiliza su directorio de trabajo actual de forma predeterminada.
 - Puede cambiar esta ruta a cualquier carpeta en la que desea almacenar las plantillas de CloudFormation.
- Seleccione su perfil de credenciales de AWS
 - Se le pedirá que elija un perfil de credenciales de AWS. El perfil seleccionado se utilizará para la detección, las validaciones y las implementaciones del entorno.
- Configure el entorno
 - Se le pedirá que cree o seleccione un entorno.
 - Los entornos definen dónde y cómo se implementarán o validarán las plantillas (por ejemplo, desarrollo, versión beta o producción). Puede usar AWS CloudFormation: CFN Init: añadir entorno para seleccionar o cambiar su entorno.
 - Puede usar AWS CloudFormation: CFN Init: eliminar entorno para eliminar el entorno que ha seleccionado.
- (Opcional) Importe archivos de parámetros
 - Si ya tiene archivos de parámetros existentes, el IDE le permite importarlos durante la inicialización.

- El IDE detecta automáticamente los archivos compatibles y los vincula a su proyecto para utilizarlos en la validación y la implementación de la plantilla.
- Asigne un nombre al proyecto y finalícelo
 - Proporcione un nombre de proyecto, como un entorno beta, y complete la configuración.
 - El IDE crea automáticamente la estructura inicial del proyecto y el archivo de configuración.

Puede ejecutar validaciones, previsualizar las implementaciones o cambiar de un entorno a otro directamente desde el IDE.

Código abierto

El servidor de lenguaje de AWS CloudFormation es de código abierto bajo la licencia de Apache-2.0, lo que brinda a los clientes total transparencia sobre cómo se realizan el diagnóstico de las plantillas, la validación de esquemas y el análisis estático. Esto reduce los problemas de seguridad y cumplimiento para los clientes que requieren visibilidad a nivel de origen antes de adoptar las herramientas.

La base de código está disponible públicamente en GitHub: <https://github.com/aws-cloudformation/cloudformation-languageserver/>.

¿Necesita ayuda?

Pruebe la [comunidad de CloudFormation](#) en AWS re:Post.

Generar plantillas de los recursos existentes con el generador IaC

Con la infraestructura de CloudFormation como generador de código (generador de IaC), puede generar una plantilla con los recursos de AWS aprovisionados en su cuenta que CloudFormation aún no administra.

Los siguientes son los beneficios del generador de IaC:

- Administre aplicaciones enteras en CloudFormation o mígrelas a una aplicación de AWS CDK.
- Genere plantillas sin tener que describir un recurso propiedad por propiedad y, a continuación, tradúzcalas a la sintaxis de JSON o YAML.
- Use la plantilla para replicar recursos en una nueva cuenta o región.

El proceso de generación de IaC consta de los siguientes pasos:

1. Escaneo de recursos: el primer paso es iniciar un escaneo de los recursos. Este escaneo se lleva a cabo en toda la región y vence a los 30 días. Durante este tiempo, puede crear varias plantillas a partir del mismo escaneo.
2. Creación de la plantilla: para crear la plantilla, tiene dos opciones:
 - Cree una plantilla nueva desde cero y agregue los recursos escaneados y los recursos relacionados.
 - Utilice una pila de CloudFormation existente como punto de partida y agregue los recursos escaneados y los recursos relacionados a su plantilla.
3. Importación de recursos: utilice la plantilla para importar los recursos como una pila de CloudFormation o para migrarlos a una aplicación de AWS CDK.

La característica de generador de IaC está disponible en todas las regiones comerciales y es compatible con muchos tipos de recursos comunes de AWS. Para obtener una lista completa de recursos admitidos, consulte [Tipo de recurso compatible](#).

Temas

- [Consideraciones](#)
- [Permisos de IAM necesarios para escanear recursos](#)
- [Comandos utilizados con frecuencia para la generación, administración y eliminación de plantillas](#)
- [Migración de una plantilla al AWS CDK](#)
- [Cómo iniciar un escaneo de recursos con el generador de IaC de CloudFormation](#)
- [Ver el resumen de análisis en la consola de CloudFormation](#)
- [Creación de una plantilla de CloudFormation a partir de recursos escaneados con el generador de IaC](#)
- [Creación de una pila de CloudFormation a partir de recursos escaneados](#)
- [Resolución de propiedades de solo escritura](#)

Consideraciones

Puede generar plantillas JSON o YAML para los recursos de AWS a los que tenga acceso de lectura. Las plantillas para la función del generador de IaC modelan los recursos en la nube de forma fiable y rápida sin tener que describir un recurso propiedad por propiedad.

En la siguiente tabla, se enumeran las cuotas disponibles para la característica de generación de laC.

Nombre	Escaneo completo	Escaneo parcial
Cantidad máxima de recursos que se pueden procesar en un escaneo	100 000	100 000
Cantidad de escaneos por día (para escaneos con menos de 10 000 recursos)	10	10
Cantidad de escaneos por día (para escaneos con más de 10 000 recursos)	1	1
Número simultáneo de plantillas generadas por cuenta	5	5
Cantidad simultánea de recursos modelados para la generación de una plantilla	5	5
Número total de recursos que se pueden modelar en una plantilla	500	500
Número máximo de plantillas generadas por cuenta	1 000	1 000

 Important

El generador de laC solo admite los recursos de AWS compatibles con la API de Cloud Control en su región. Para obtener más información, consulte [Tipo de recurso compatible](#).

Permisos de IAM necesarios para escanear recursos

Para escanear los recursos con el generador de laC, su entidad principal de IAM (usuario, rol o grupo) debe tener:

- Permisos de escaneo de CloudFormation
- Permisos de lectura para los servicios de destino de AWS

El alcance del escaneo se limita a los recursos a los que tiene acceso de lectura. Si faltan permisos, el escaneo no generará un error, pero excluirá esos recursos.

Para ver una política de IAM de ejemplo que concede permisos de escaneo y de gestión de plantillas, consulte [Permitir todas las operaciones del generador de IaC](#).

Comandos utilizados con frecuencia para la generación, administración y eliminación de plantillas

Los comandos utilizados con frecuencia para trabajar con políticas de escalado incluyen los siguientes:

- [start-resource-scan](#) para iniciar un escaneo de los recursos de la cuenta en una Región de AWS.
- [describe-resource-scan](#) para supervisar el progreso de un escaneo de recursos.
- [list-resource-scans](#) para enumerar los escaneos de recursos en una Región de AWS.
- [list-resource-scan-resources](#) para enumerar los recursos que se encuentran durante el escaneo de recursos.
- [list-resource-scan-related-resources](#) para enumerar los recursos relacionados con los recursos escaneados.
- [create-generated-template](#) para generar una plantilla de CloudFormation a partir de un conjunto de recursos escaneados.
- [update-generated-template](#) para actualizar la plantilla generada.
- [describe-generated-template](#) para devolver información sobre una plantilla generada.
- [list-generated-templates](#) para enumerar todas las plantillas generadas en su cuenta y región actual.
- [delete-generated-template](#) para eliminar una plantilla generada.

Migración de una plantilla al AWS CDK

El AWS Cloud Development Kit (AWS CDK) es un marco de desarrollo de software de código abierto que puede utilizar para desarrollar, administrar e implementar los recursos de CloudFormation mediante lenguajes de programación populares.

La CLI de AWS CDK proporciona una integración con el generador de IaC. Use el comando `cdk migrate` de la CLI del AWS CDK para generar una plantilla de CloudFormation y crear una nueva

aplicación del CDK que contenga los recursos. A continuación, puede utilizar el AWS CDK para administrar los recursos y las implementaciones para CloudFormation.

Para obtener más información, consulte [Migrar a AWS CDK](#) en la Guía para desarrolladores de AWS Cloud Development Kit (AWS CDK).

Cómo iniciar un escaneo de recursos con el generador de laC de CloudFormation

Antes de crear una plantilla a partir de recursos existentes, primero debe iniciar un escaneo de recursos para descubrir los recursos actuales y sus relaciones.

Puede iniciar un escaneo de recursos a través de una de las siguientes opciones. Recomendamos la primera opción para quienes utilizan el generador de laC por primera vez.

- Escanear todos los recursos (escaneo completo): escanea todos los recursos existentes en la cuenta y región actuales. Este proceso de escaneo puede tardar hasta 10 minutos para 1000 recursos.
- Escanear recursos específicos (escaneo parcial): seleccione manualmente qué tipos de recursos escanear en la cuenta y la región actuales. Esta opción ofrece un proceso de escaneo más rápido y enfocado, lo que la hace ideal para el desarrollo iterativo de plantillas.

Una vez que se completa el escaneo, puede elegir qué recursos y sus recursos relacionados incluir al generar la plantilla. Al utilizar el escaneo parcial, los recursos relacionados solo estarán disponibles durante la generación de la plantilla si se cumple alguna de las siguientes condiciones:

- Los seleccionó específicamente antes de iniciar el escaneo, o
- Fueron necesarios para detectar los tipos de recursos que seleccionó.

Por ejemplo, si selecciona un `AWS::EKS::Nodegroup` sin seleccionar el `AWS::EKS::Cluster`, el generador de laC incluye automáticamente los recursos del `AWS::EKS::Cluster` en el escaneo, ya que para detectar el grupo de nodos es necesario detectar primero el clúster. En todos los demás casos, el escaneo solo incluirá los recursos que seleccione específicamente.

 Note

Antes de continuar, confirme que cuenta con los permisos necesarios para trabajar con el generador de laC. Para obtener más información, consulte [Permisos de IAM necesarios para escanear recursos](#).

Temas

- [Inicio de un escaneo de recursos \(consola\)](#)
- [Inicio de un escaneo de recursos \(AWS CLI\)](#)

Inicio de un escaneo de recursos (consola)

Para iniciar un escaneo de recursos de todos los tipos de recursos (escaneo completo)

1. Abra la [página del generador de laC](#) de la consola de CloudFormation.
2. En la barra de navegación de la parte superior de la pantalla, elija la Región de AWS que contenga los recursos que desea escanear.
3. En el panel Escaneos, elija Iniciar un nuevo escaneo y, a continuación, elija Escanear todos los recursos.

Para iniciar un escaneo de recursos de tipos de recursos específicos (escaneo parcial)

1. Abra la [página del generador de laC](#) de la consola de CloudFormation.
2. En la barra de navegación de la parte superior de la pantalla, elija la Región de AWS que contenga los recursos que desea escanear.
3. En el panel Escaneos, elija Iniciar un nuevo escaneo y, a continuación, elija Escanear recursos específicos.
4. En el cuadro de diálogo Iniciar escaneo parcial, seleccione hasta 100 tipos de recursos y, a continuación, elija Iniciar escaneo.

Inicio de un escaneo de recursos (AWS CLI)

Para iniciar un escaneo de recursos de todos los tipos de recursos (escaneo completo)

Utilice el siguiente comando [start-resource-scan](#). Sustituya *us-east-1* por la Región de AWS que contenga los recursos que desea escanear.

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

Si se ejecuta correctamente, este comando devolverá el ARN del escaneo. Tenga en cuenta el ARN de la propiedad ResourceScanId. Lo necesita para crear la plantilla.

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

Para iniciar un escaneo de recursos de tipos de recursos específicos (escaneo parcial)

1. Use el comando [cat](#) para almacenar los tipos de recursos que desea escanear en un archivo JSON con el nombre `config.json` en el directorio principal. A continuación se muestra un ejemplo de configuración de escaneo que busca instancias de Amazon EC2, grupos de seguridad y todos los recursos de Amazon S3.

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

2. Utilice el comando [start-resource-scan](#) con la opción `--scan-filters`, junto con el archivo `config.json` que creó, para iniciar el análisis parcial. Sustituya *us-east-1* por la Región de AWS que contenga los recursos que desea escanear.

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

Si se ejecuta correctamente, este comando devolverá el ARN del escaneo. Tenga en cuenta el ARN de la propiedad `ResourceScanId`. Lo necesita para crear la plantilla.

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

Para monitorear el progreso de un análisis de recursos

Utilice el comando [describe-resource-scan](#). Para la opción `--resource-scan-id`, reemplace el ARN de muestra por el ARN real.

```
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
```

Si se ejecuta correctamente, este comando proporciona información similar a la siguiente:

```
{
  "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
}

```

Así luce la salida generada por un escaneo parcial:

```

{
  "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::*"
      ]
    }
  ]
}

```

Para obtener una descripción de los campos de la salida, consulte [DescribeResourceScan](#) en la Referencia de la API de AWS CloudFormation.

Ver el resumen de análisis en la consola de CloudFormation

Una vez finalizado el análisis, puede ver una visualización de los recursos encontrados durante el análisis para ayudarse a identificar la concentración de recursos en los diferentes tipos de productos.

Para ver la información de los recursos encontrados durante el análisis

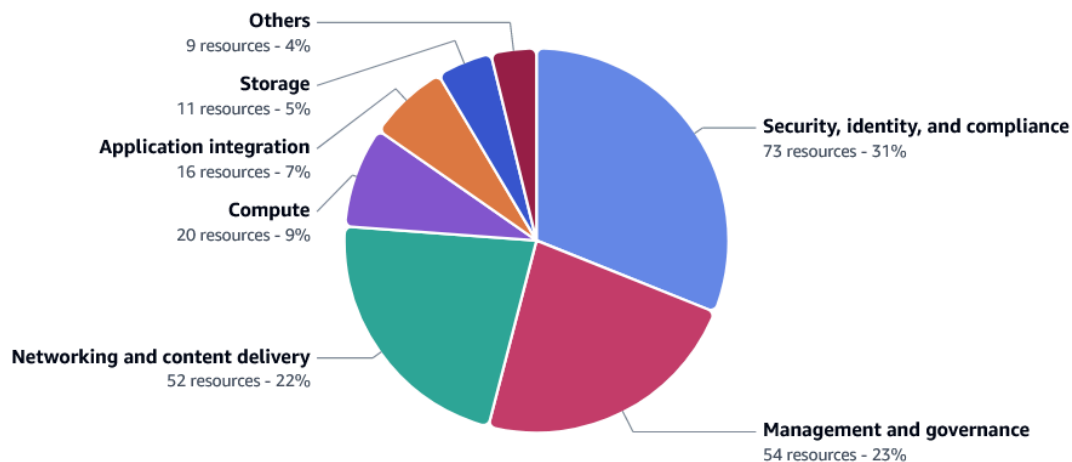
1. Abra la [página del generador de laC](#) de la consola de CloudFormation.
2. En la barra de navegación de la parte superior de la pantalla, elija la Región de AWS que contenga los recursos que desea escanear para ver.
3. Elija Generador de laC en el panel de navegación.
4. En el Desglose de recursos analizados, encontrará un desglose visual de los recursos escaneados por tipo de producto, por ejemplo, Computación y Almacenamiento.
5. Para personalizar la cantidad de tipos de productos que se muestran, seleccione Filtrar los datos mostrados. Esto ayuda a personalizar la visualización para centrarse en los tipos de productos que más le interesan.
6. En el lado derecho de la página está el panel Setalles del resumen de análisis. Para abrir el panel, seleccione el icono abrir panel.

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 ▼



Product type

Security, identity, and compliance Management and governance Networking and content delivery Compute Application integration Storage Others

Creación de una plantilla de CloudFormation a partir de recursos escaneados con el generador de IaC

En este tema se explica cómo crear una plantilla a partir de los recursos escaneados con la característica de generación de IaC.

Creación de una plantilla a partir de recursos escaneados (consola)

Creación de una plantilla de pila a partir de recursos escaneados

1. Abra la [página del generador de IaC](#) de la consola de CloudFormation.
2. En la barra de navegación de la parte superior de la pantalla, elija la Región de AWS que contenga los recursos escaneados.
3. En la sección Plantillas, elija Crear plantilla.
4. Elija Empezar desde una plantilla nueva.
 - a. En Nombre de plantilla, proporcione un nombre para la plantilla.
 - b. (Opcional) Configure la Política de eliminación y Actualice la política de reemplazo.
 - c. Seleccione Siguiente para añadir los recursos escaneados a la plantilla.
5. En Agregar recursos analizados, navegue por la lista de recursos escaneados y seleccione los recursos que desee agregar a la plantilla. Puede filtrar los recursos por identificador de recurso, tipo de recurso o etiquetas. Los filtros se incluyen entre sí.
6. Cuando haya agregado todos los recursos necesarios a la plantilla, elija Siguiente para salir de la página Agregar recursos analizados y pasar a la página Agregar recursos relacionados.
7. Revise una lista recomendada de recursos relacionados. Los recursos relacionados, como las instancias de Amazon EC2 y los grupos de seguridad, son interdependientes y, por lo general, pertenecen a la misma carga de trabajo. Seleccione los recursos relacionados que desee incluir en la plantilla de generada.

Note

Le sugerimos que añada todos los recursos relacionados a esta plantilla.

8. Revise los detalles de la plantilla, los recursos escaneados y los recursos relacionados.
9. Seleccione Crear plantilla para salir de la página Revisar y crear y crear la plantilla.

Creación de una plantilla a partir de recursos escaneados (AWS CLI)

Creación de una plantilla de pila a partir de recursos escaneados

1. Para enumerar los recursos encontrados durante el escaneo, use el comando [list-resource-scan-resources](#) y, si lo desea, especifique la opción `--resource-identifier` para limitar las salidas. Para la opción `--resource-scan-id`, reemplace el ARN de muestra por el ARN real.

```
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
```

A continuación, se muestra una respuesta de ejemplo, en la que `ManagedByStack` indica si CloudFormation ya administra el recurso. Copie la salida. Lo necesita para el siguiente paso.

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

Para obtener una descripción de los campos de la salida, consulte [ScannedResource](#) en la Referencia de la API de AWS CloudFormation.

2. Use el comando `cat` para almacenar los tipos de recursos y los identificadores en un archivo JSON con el nombre `resources.json` en su directorio principal. El siguiente es un JSON de ejemplo que se basa en la salida del ejemplo del paso anterior.

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

3. Para enumerar los recursos encontrados durante el escaneo, use el comando [list-resource-scan-related-resources](#), junto con el archivo `resources.json` que creó.

```
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
```

A continuación, se muestra una respuesta de ejemplo, en la que `ManagedByStack` indica si CloudFormation ya administra el recurso. Agregue estos recursos al archivo JSON que creó en el paso anterior. Lo necesitará para crear la plantilla.

```
{
  "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
    }
]
}

```

Para obtener una descripción de los campos de la salida, consulte [ScannedResource](#) en la Referencia de la API de AWS CloudFormation.

Note

La lista de recursos de entrada no puede superar una longitud de 100. Para enumerar los recursos relacionados de más de 100 recursos, ejecute el comando `list-resource-scan-related-resources` en lotes de 100 y consolide los resultados.

Tenga en cuenta que el resultado puede contener recursos duplicados en la lista.

4. Para crear una plantilla de pila nueva, use el comando [create-generated-template](#) como se muestra a continuación y con estas modificaciones:

- Sustituya *us-east-1* por la Región de AWS que contenga los recursos escaneados.
- Sustituya *MyTemplate* por el nombre de la plantilla a crear.

```

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

```

A continuación se muestra un ejemplo de un archivo `resources.json`.

```

[
  {
    "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"
      }
    }
  ]
}
```

Si se ejecuta correctamente, este comando devuelve lo siguiente.

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

Creación de una pila de CloudFormation a partir de recursos escaneados

Después de crear la plantilla, puede obtener una vista previa de la plantilla generada con Infrastructure Composer antes de crear la pila e importar los recursos escaneados. Esto ayuda a visualizar la arquitectura completa de la aplicación con los recursos y sus relaciones. Para obtener más información acerca de Infrastructure Composer, consulte [Creación de plantillas visualmente con Infrastructure Composer](#).

Creación de la pila e importación de los recursos escaneados

1. Abra la [página del generador de laC](#) de la consola de CloudFormation.

2. En la barra de navegación de la parte superior de la pantalla, elija la Región de AWS para la plantilla.
3. Seleccione la pestaña Plantillas y, a continuación, elija el nombre de la plantilla para ver más información.
4. En la pestaña Definición de plantillas, en la parte superior de la sección Plantillas, puede cambiar la plantilla de la sintaxis YAML a la JSON según sus preferencias.
5. Revise los detalles de la plantilla para confirmar que todo está configurado correctamente. Para facilitar la revisión y la comprensión de la plantilla, puede cambiar de la vista de código predeterminada a una vista gráfica de la infraestructura descrita en la plantilla mediante Infrastructure Composer. Para ello, en Plantilla, elija Lienzo en lugar de Plantilla.

Acciones del lienzo

- Para centrarse en los detalles de un recurso específico de su plantilla, haga doble clic en una tarjeta para que aparezca el panel Propiedades del recurso.
 - Para disponer y organizar visualmente las tarjetas en el lienzo, seleccione Organizar en la parte superior izquierda del lienzo.
 - Para acercar y alejar el lienzo, usa los controles de zoom de la parte inferior derecha del lienzo.
6. Para ver un recurso específico en la consola, seleccione la pestaña Recursos de plantilla y, a continuación, elija el ID físico del recurso que desea consultar. Esto le lleva a la consola de ese recurso específico. También puede añadir, eliminar y resincronizar los recursos de la definición de plantilla desde la pestaña Recursos de plantilla.
 7. En la pestaña Definición de plantilla, el generador de IaC puede emitir advertencias sobre los recursos que contienen propiedades de solo escritura. Tras revisar las advertencias, puede descargar la plantilla generada y hacer los cambios necesarios. Para obtener más información, consulte [Resolución de propiedades de solo escritura](#).
 8. Cuando esté satisfecho con la definición de la plantilla, en la plantilla Definición de plantilla, elija Importar a la pila y, a continuación, elija Siguiente.
 9. En el panel Especificar pila de la página Especificar los detalles de la pila, ingrese el nombre de la pila y, a continuación, elija Siguiente.
 10. Revise e introduzca los parámetros de la pila. Elija Siguiente.
 11. Revise sus opciones en la página Revisar cambios y seleccione Siguiente.
 12. Revise los detalles en la página Revisar e importar y seleccione Importar recursos.

Resolución de propiedades de solo escritura

Con el generador de IaC de CloudFormation, puede generar una plantilla con los recursos aprovisionados en su cuenta que CloudFormation aún no administra. Sin embargo, algunas propiedades de recursos se designan como de solo escritura, lo que significa que CloudFormation puede escribirlas, pero no leerlas; por ejemplo, una contraseña de base de datos.

Al generar plantillas de CloudFormation a partir de los recursos existentes, las propiedades de solo escritura representan un desafío. En la mayoría de los casos, CloudFormation convierte estas propiedades en parámetros en la plantilla generada. Esto le permite introducir las propiedades como valores de parámetros durante las operaciones de importación. Sin embargo, hay situaciones en las que esta conversión no es posible y CloudFormation gestiona estos casos de forma diferente.

Propiedades que se excluyen mutuamente

Algunos recursos tienen varios conjuntos de propiedades exclusivos y al menos algunos de estos son de solo escritura. El generador de IaC no puede determinar qué conjunto de propiedades exclusivas se aplicó al recurso durante la creación. Por ejemplo, puede proporcionar el código para [AWS::Lambda::Function](#) al utilizar uno de estos conjuntos de propiedades.

- Code/S3Bucket, Code/S3Key, y en forma opcional Code/S3ObjectVersion
- Code/ImageUri
- Code/ZipFile

Todas estas propiedades son de solo escritura. El generador de IaC selecciona uno de los conjuntos exclusivos de propiedades y lo añade a la plantilla generada. Se añaden parámetros para cada una de las propiedades de solo escritura. Los nombres de los parámetros incluyen `OneOf` y las descripciones de los parámetros indican que la propiedad correspondiente se puede reemplazar por otras propiedades exclusivas. El generador de IaC establece un tipo de advertencia de `MUTUALLY_EXCLUSIVE_PROPERTIES` para las propiedades incluidas.

Tipos que se excluyen mutuamente

En algunos casos, una propiedad de solo escritura puede ser de varios tipos de datos. Por ejemplo, la propiedad `Body` de [AWS::ApiGateway::RestApi](#) puede ser un `object` o una `string`. Cuando este es el caso, el generador de IaC incluye la propiedad en la plantilla generada mediante el tipo de `string` y establece un tipo de advertencia de `MUTUALLY_EXCLUSIVE_TYPES`.

Propiedades de Array

Si una propiedad de solo escritura tiene un tipo de `array`, el generador de laC no puede incluirla en la plantilla generada porque los parámetros solo pueden ser valores escalares. En este caso, la propiedad se omite en la plantilla y se establece un tipo de advertencia de `UNSUPPORTED_PROPERTIES`.

Propiedades opcionales

En el caso de las propiedades de solo escritura opcionales, el generador de laC no puede detectar si la propiedad se utilizó al configurar el recurso. En este caso, la propiedad se omite de la plantilla generada y se establece un tipo de advertencia de `UNSUPPORTED_PROPERTIES`.

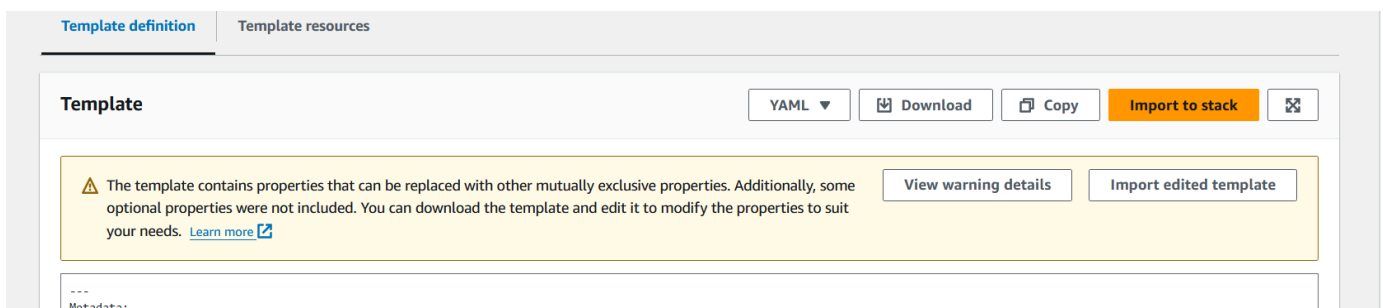
Advertencias y próximos pasos

Para determinar qué propiedades son de solo escritura, debe observar las advertencias devueltas por la consola del generador de laC. La [Referencia de tipos de recursos y propiedades de AWS](#) no indica si una propiedad es de solo escritura o si admite varios tipos.

Como alternativa, puede ver qué propiedades son de solo escritura desde los esquemas del proveedor de recursos. Para descargar los esquemas del proveedor de recursos, consulte los [Esquemas de proveedores de recursos de CloudFormation](#).

Resolución de problemas relacionados con las propiedades de solo

1. Abra la [página del generador de laC](#) de la consola de CloudFormation.
2. En la barra de navegación de la parte superior de la pantalla, elija la Región de AWS para la plantilla.
3. Seleccione la pestaña Plantillas y, a continuación, elija el nombre de la plantilla que creó.
4. En la pestaña Definición de plantilla cuando la plantilla generada incluye recursos con propiedades de solo escritura, la consola del generador de laC muestra una advertencia con un resumen del tipo de problema. Por ejemplo:



5. Elija Ver detalles de la advertencia para conocer más detalles. Los recursos con propiedades de solo escritura se identifican mediante el ID lógico utilizado en la plantilla generada y en el tipo de recurso.

Utilice la lista de advertencias para identificar los recursos con propiedades de solo escritura y observe cada recurso para determinar qué cambios (si los hay) deben realizarse en la plantilla generada.

Resource warnings (1) ×

▼ **LambdaFunction00laClambda002fd6K (AWS::Lambda::Function)** [Learn more](#)

Unsupported properties	Mutually exclusive properties
1. SnapStart/ApplyOn Applying SnapStart setting on function resource type. Possible values: [PublishedVersions, None] This property can be replaced with other types 2. Code/S3ObjectVersion For versioned objects, the version of the deployment package object to use. This property can be replaced with other 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. Si la plantilla debe actualizarse para resolver problemas con las propiedades de solo escritura, haga lo siguiente:
 - a. Seleccione Descargar para descargar una copia de la plantilla.
 - b. Edite la plantilla.
 - c. Una vez finalizados los cambios, puede pulsar el botón Importar plantilla editada para continuar con el proceso de importación.

Resolución de problemas relacionados con las propiedades de solo escritura en recursos de AWS::ApiGateway::RestAPI

En este tema se explica cómo resolver los problemas relacionados con las propiedades de solo escritura de los recursos de [AWS::ApiGateway::RestApi](#) cuando se utiliza el generador de laC.

Problema

Cuando una plantilla generada contiene recursos `AWS::ApiGateway::RestApi`, se generan advertencias que indican que las propiedades de `Body`, `BodyS3Location` y `CloneFrom` se identifican como `UNSUPPORTED_PROPERTIES`. Esto se debe a que son propiedades de solo escritura opcionales. El generador de laC no sabe si estas propiedades se aplicaron alguna vez al recurso. Por lo tanto, omita estas propiedades en la plantilla generada.

Resolución

Para establecer la propiedad Body de la API de REST, actualice la plantilla generada.

1. Utilice la acción de la API de [GetExport](#) de Amazon API Gateway para descargar la API. Por ejemplo, al utilizar el comando [aws apigateway get-export](#) de la AWS CLI. Para obtener más información, consulte [Exportar una API de REST desde API Gateway](#) en la Guía para desarrolladores de API Gateway.
2. Recupere la propiedad Body de la respuesta a la acción de la API de GetExport. Carguela en un bucket de Amazon S3.
3. Descargue la plantilla generada.
4. Añada las propiedades de BodyS3Location/Bucket y BodyS3Location/Key a la plantilla y especifique el nombre y la clave del bucket donde se almacena Body.
5. Abra la plantilla generada en la consola del generador de IaC y seleccione Importar plantilla editada.

Resolución de problemas relacionados con las propiedades de solo escritura en recursos de AWS::Lambda::Function

En este tema se explica cómo resolver los problemas relacionados con las propiedades de solo escritura de los recursos de [AWS::Lambda::Function](#) cuando se utiliza el generador de IaC.

Problema

El recurso de `AWS::Lambda::Function` tiene tres conjuntos de propiedades que se excluyen mutuamente para especificar el código de Lambda:

- Propiedades `Code/S3Bucket` y `Code/S3Key` y, de manera opcional, la propiedad `Code/S3ObjectVersion`
- Propiedad `Code/ImageUri`
- Propiedad `Code/ZipFile`

Solo se puede usar uno de estos conjuntos para un recurso de `AWS::Lambda::Function` determinado.

El generador de IaC no puede determinar qué conjunto de propiedades exclusivas de solo escritura se utilizó para crear o actualizar el recurso. Como resultado, incluye solo el primer conjunto de

propiedades de la plantilla generada. Se omiten las propiedades `Code/ImageUri` y `Code/ZipFile`.

Además, el generador de laC emite las siguientes advertencias:

- **MUTUALLY_EXCLUSIVE_PROPERTIES**: advierte que `Code/S3Bucket` y `Code/S3Key` se identifican como propiedades que se excluyen mutuamente.
- **UNSUPPORTED_PROPERTIES**: advierte que la propiedad `Code/S3ObjectVersion` no es compatible.

Para incluir recursos de `AWS::Lambda::Function` en una plantilla generada, debe descargar y actualizar la plantilla con las propiedades de código correctas.

Resolución

Si almacena el código Lambda en un bucket de Amazon S3 y no utiliza la propiedad **`S3ObjectVersion`**, puede importar la plantilla generada sin ninguna modificación. El generador de laC le pedirá el bucket de Amazon S3 y la clave como parámetros de plantilla durante la operación de importación.

Si almacena el código Lambda como un repositorio de Amazon ECR, puede actualizar la plantilla de acuerdo con las siguientes instrucciones:

1. Descargue la plantilla generada.
2. Elimine las propiedades y los parámetros correspondientes de las propiedades de `Code/S3Bucket` y `Code/S3Key` de la plantilla generada.
3. Sustituya las propiedades eliminadas de la plantilla generada por la propiedad `Code/ImageUri` y especifique la URL del repositorio de Amazon ECR.
4. Abra la plantilla generada en la consola del generador de laC y pulse el botón Importar plantilla editada.

Si almacena el código Lambda en un archivo zip, puede actualizar la plantilla de acuerdo con las siguientes instrucciones:

1. Descargue la plantilla generada.
2. Elimine las propiedades y los parámetros correspondientes de las propiedades de `Code/S3Bucket` y `Code/S3Key` de la plantilla generada.

3. Sustituya las propiedades eliminadas de la plantilla generada por la propiedad `Code/ZipFile`.
4. Abra la plantilla generada en la consola del generador de laC y pulse el botón Importar plantilla editada.

Si no tiene una copia del código Lambda, puede actualizar la plantilla según las siguientes instrucciones:

1. Utilice la acción de la API de AWS Lambda [GetFunction](#) (por ejemplo, mediante el comando [aws lambda get-function](#) de la AWS CLI).
2. En la respuesta, el parámetro `RepositoryType` indica S3 si el código está en un bucket de Amazon S3 o ECR si el código se encuentra en un repositorio de Amazon ECR.
3. En la respuesta, el parámetro `Location` contiene una URL prefirmada que puede usar para descargar el paquete de implementación durante 10 minutos. Descargue el código.
4. Cargue el archivo en un bucket de Amazon S3.
5. Ejecute una operación de importación con la plantilla generada y proporcione el nombre y la clave del bucket como valores de parámetro.

Obtención de valores almacenados en otros servicios con referencias dinámicas

Las referencias dinámicas ofrecen una forma cómoda de especificar valores externos almacenados y administrados en otros servicios y de separar la información confidencial de sus plantillas de infraestructura como código. CloudFormation recupera el valor de la referencia especificada cuando es necesario durante las operaciones de pila y de conjunto de cambios.

Con las referencias dinámicas, puede:

- Usar cadenas seguras: en el caso de los datos confidenciales, use siempre parámetros de cadenas seguras en el Almacén de datos de AWS Systems Manager o secretos en AWS Secrets Manager para garantizar que sus datos estén cifrados en reposo.
- Limitar el acceso: restrinja el acceso a los parámetros del Almacén de parámetros o a los secretos de Secrets Manager solo a las entidades principales y roles autorizados.
- Rotar las credenciales: rote periódicamente los datos confidenciales almacenados en el Almacén de parámetros o Secrets Manager para mantener un alto nivel de seguridad.

- Automatizar la rotación: aproveche la característica de rotación automática de Secrets Manager para actualizar y distribuir periódicamente sus datos confidenciales en sus aplicaciones y entornos.

Consideraciones generales

A continuación, se ofrecen consideraciones generales para que tenga en cuenta antes de especificar referencias dinámicas en sus plantillas de CloudFormation:

- No incluya referencias dinámicas ni ningún dato confidencial en las propiedades de los recursos que formen parte del identificador principal de un recurso. CloudFormation puede usar el valor real de texto sin formato en el identificador del recurso principal, lo que podría suponer un riesgo para la seguridad. Este ID de recurso puede aparecer en cualquier destino o salida derivada.

Para determinar qué propiedades de recursos comprenden el identificador principal de un tipo de recurso, consulte la documentación de referencia de recursos para ese recurso en [Referencia de tipos de recursos y propiedades de AWS](#). En la sección Return values (Valores de devolución), el valor de devolución de la función Ref representa las propiedades del recurso que comprenden el identificador principal del tipo de recurso.

- Puede incluir hasta 60 referencias dinámicas en una plantilla de pila.
- Si utiliza transformaciones (como `AWS::Include` o `AWS::Serverless`), CloudFormation no resuelve las referencias dinámicas antes de aplicar la transformación. En su lugar, pasa la cadena literal de la referencia dinámica a la transformación y resuelve las referencias al ejecutar el conjunto de cambios usando la plantilla.
- No se pueden utilizar referencias dinámicas para valores protegidos (como los almacenados en el Almacén de parámetros o Secrets Manager) en recursos personalizados.
- Las referencias dinámicas tampoco se admiten en los metadatos de `AWS::CloudFormation::Init` ni en las propiedades de `UserData` de Amazon EC2.
- No cree una referencia dinámica que termine con una barra invertida (`\`). CloudFormation no puede resolver estas referencias, lo que provocará un error en las operaciones de apilado.

Los siguientes temas brindan información y otras consideraciones sobre el uso de referencias dinámicas.

Temas

- [Obtención de un valor de texto sin formato del Almacén de parámetros de Systems Manager](#)
- [Obtención de un valor de cadena segura del Almacén de parámetros de Systems Manager](#)

- [Obtención de un secreto o un valor secreto de Secrets Manager](#)

Obtención de un valor de texto sin formato del Almacén de parámetros de Systems Manager

Al crear una plantilla de CloudFormation, es posible que desee utilizar valores de texto sin formato almacenados en el Almacén de parámetros. Parameter Store es una función de AWS Systems Manager. Para obtener una introducción al Almacén de parámetros, consulte [Almacén de parámetros de AWS Systems Manager](#) en la Guía del usuario de AWS Systems Manager.

Para utilizar un valor de texto sin formato del Almacén de parámetros en la plantilla, debe usar una referencia dinámica ssm. Esta referencia permite acceder a valores de parámetros de tipo String o StringList del Almacén de parámetros.

Para verificar qué versión de una referencia dinámica ssm se utilizará en una operación de pila, cree un conjunto de cambios para la operación de pila. A continuación, revise la plantilla procesada en la pestaña Plantilla. Para obtener más información, consulte [Creación de un conjunto de cambios para una pila de CloudFormation](#).

Al utilizar referencias dinámicas ssm, debe tener en cuenta varios aspectos:

- CloudFormation no es compatible con la detección de desviaciones en referencias dinámicas. Para las referencias dinámicas de ssm en las que no se ha especificado la versión del parámetro, se recomienda que, si actualiza la versión del parámetro en Systems Manager, también lleve a cabo una operación de actualización de la pila en cualquier pila que incluya la referencia dinámica de ssm, con el fin de obtener la versión más reciente del parámetro.
- Para usar una referencia dinámica ssm en la sección `Parameters` de la plantilla de CloudFormation, debe incluir un número de versión. CloudFormation no permite referenciar un valor del Almacén de parámetros sin un número de versión en esta sección. Como alternativa, puede definir el parámetro como un tipo de parámetro de Systems Manager en la plantilla. Al hacerlo, puede especificar una clave de parámetro de Systems Manager como valor predeterminado. A continuación, CloudFormation obtendrá la última versión del valor del parámetro del Almacén de parámetros, sin que tenga que especificar un número de versión. Esto puede hacer que sus plantillas sean más sencillas y fáciles de mantener. Para obtener más información, consulte [Definición de recursos existentes en tiempo de ejecución con tipos de parámetros proporcionados por CloudFormation](#).

- En el caso de recursos personalizados, CloudFormation resuelve las referencias dinámicas ssm antes de enviar la solicitud al recurso personalizado.
- CloudFormation no admite el uso de referencias dinámicas para hacer referencia a un parámetro compartido desde otra Cuenta de AWS.
- CloudFormation no admite el uso de etiquetas de parámetros de Systems Manager en las referencias dinámicas.

Permisos

Para especificar un parámetro guardado en el Almacén de parámetros de Systems Manager, debe tener permiso para llamar a [GetParameters](#) para el parámetro especificado. Para obtener información sobre cómo crear políticas de IAM que proporcionen acceso a parámetros específicos de Systems Manager, consulte [Restricción del acceso a los parámetros de Systems Manager mediante políticas de IAM](#) en la Guía del usuario de AWS Systems Manager.

Patrón de referencia

Para referenciar un valor de texto sin formato guardado en el Almacén de parámetros de Systems Manager en una plantilla de CloudFormation, utilice el siguiente patrón de referencia ssm.

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

Su referencia debe cumplir el siguiente patrón de expresión regular para nombre-parámetro y versión:

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

parameter-name

El nombre del parámetro en el almacén de parámetros de . El nombre de parámetro distingue entre mayúsculas y minúsculas.

Obligatorio.

version

Un número entero que especifica la versión del parámetro que se va a utilizar. Si no especifica la versión exacta, CloudFormation utiliza la última versión del parámetro cada vez que cree o

actualice la pila. Para obtener más información, consulte [Trabajo con versiones de parámetros](#) en la Guía del usuario de AWS Systems Manager.

Opcional.

Ejemplos

Temas

- [Parámetro de ID de AMI públicas](#)
- [Parámetro de ID de AMI personalizado](#)

Parámetro de ID de AMI públicas

En el siguiente ejemplo, se crea una instancia de EC2 que refiere a un parámetro de AMI público. La referencia dinámica recupera el ID de AMI de Amazon Linux 2023 del parámetro público. Para obtener más información acerca de parámetros públicos, consulte [Descubrir parámetros públicos en el Almacén de parámetros](#) en la Guía del usuario de AWS Systems Manager.

JSON

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

YAML

```
Resources:
  MyInstance:
    Type: AWS::EC2::Instance
    Properties:
```

```
ImageId: '{{resolve:ssm:/aws/service/ami-amazon-linux-latest/al2023-ami-
kernel-6.1-x86_64}}'
InstanceType: t2.micro
```

Parámetro de ID de AMI personalizado

En el siguiente ejemplo, se crea una plantilla de lanzamiento de EC2 que refiere a un ID de AMI personalizado almacenado en el Almacén de parámetros. La referencia dinámica recupera el ID de AMI de la versión **2** del parámetro *golden-ami* en el momento en que se lanza una instancia desde la plantilla de lanzamiento.

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
```

Obtención de un valor de cadena segura del Almacén de parámetros de Systems Manager

En CloudFormation, puede usar datos confidenciales, como contraseñas o claves de licencia, sin exponerlos directamente en sus plantillas al almacenar los datos confidenciales como una “cadena segura” en el Almacén de parámetros de AWS Systems Manager. Para obtener una introducción al Almacén de parámetros, consulte [Almacén de parámetros de AWS Systems Manager](#) en la Guía del usuario de AWS Systems Manager.

Para utilizar una cadena segura del Almacén de parámetros en la plantilla, debe usar una referencia dinámica `ssm-secure`. CloudFormation no almacena nunca el valor real de cadena segura. En su lugar, solo almacena la referencia dinámica literal, que contiene el nombre de parámetro en texto sin formato de la cadena segura.

Durante la creación o las actualizaciones de la pila, CloudFormation accede al valor de cadena segura según sea necesario, sin exponer el valor real. Las cadenas seguras solo se pueden usar para las propiedades de recursos que admiten el patrón de referencia dinámica `ssm-secure`. Para obtener más información, consulte [Recursos que admiten patrones de parámetros dinámicos para SecureStrings o cadenas seguras](#).

CloudFormation no devuelve el valor del parámetro real para las cadenas seguras en ninguna llamada a la API. Solo devuelve la referencia dinámica literal. Al comparar los cambios con el conjunto de cambios, CloudFormation solo compara la cadena de referencia dinámica literal. No resuelve ni compara los valores reales de la cadena segura.

Al utilizar referencias dinámicas `ssm-secure`, debe tener en cuenta varios aspectos:

- CloudFormation no puede acceder a los valores del Almacén de parámetros desde otras Cuentas de AWS.
- CloudFormation no es compatible con el uso de etiquetas de parámetros de Systems Manager o parámetros públicos en las referencias dinámicas.
- En las regiones `cn-north-1` y `cn-northwest-1`, las cadenas seguras no son compatibles con Systems Manager.
- Las referencias dinámicas de valores seguros, como, por ejemplo, `ssm-secure`, no son compatibles actualmente con los recursos personalizados.
- Si CloudFormation necesita reversionar la actualización de la pila, y la versión especificada anteriormente de un parámetro de cadena segura ya no está disponible, se producirá un error en dicha operación de reversión. En esos casos, existen dos opciones:

- Use `CONTINUE_UPDATE_ROLLBACK` para omitir el recurso.
- Vuelva a crear el parámetro `SecureString` o de cadena segura en el almacén de parámetros de Administrador de sistemas y actualícelo hasta que la versión del parámetro llegue a la versión que se utiliza en la plantilla. Luego use `CONTINUE_UPDATE_ROLLBACK` sin omitir el recurso.

Recursos que admiten patrones de parámetros dinámicos para `SecureStrings` o cadenas seguras

Los recursos compatibles con el patrón de referencia dinámica `ssm-secure` son:

Recurso	Tipo de propiedad	Propiedades
<code>AWS::DirectoryService::MicrosoftAD</code>		Password
<code>AWS::DirectoryService::SimpleAD</code>		Password
<code>AWS::ElastiCache::ReplicationGroup</code>		AuthToken
<code>AWS::IAM::User</code>	<code>LoginProfile</code>	Password
<code>AWS::KinesisFirehose::DeliveryStream</code>	<code>RedshiftDestinationConfiguration</code>	Password
<code>AWS::OpsWorks::App</code>	<code>Origen</code>	Password
<code>AWS::OpsWorks::Stack</code>	<code>CustomCookbooksSource</code>	Password
<code>AWS::OpsWorks::Stack</code>	<code>RdsDbInstances</code>	DbPassword
<code>AWS::RDS::DBCluster</code>		MasterUserPassword
<code>AWS::RDS::DBInstance</code>		MasterUserPassword
<code>AWS::Redshift::Cluster</code>		MasterUserPassword

Patrón de referencia

Para referenciar un valor de cadena segura del Almacén de parámetros de Systems Manager en una plantilla de CloudFormation, utilice el siguiente patrón de referencia `ssm-secure`.

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

Su referencia debe cumplir el siguiente patrón de expresión regular para nombre-parámetro y versión:

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

parameter-name

El nombre del parámetro en el almacén de parámetros de . El nombre de parámetro distingue entre mayúsculas y minúsculas.

Obligatorio.

version

Un número entero que especifica la versión del parámetro que se va a utilizar. Si no especifica la versión exacta, CloudFormation utiliza la última versión del parámetro cada vez que cree o actualice la pila. Para obtener más información, consulte [Trabajo con versiones de parámetros](#) en la Guía del usuario de AWS Systems Manager.

Opcional.

Ejemplo

En el siguiente ejemplo, se utiliza una referencia dinámica `ssm-secure` para establecer la contraseña de un usuario de IAM para una cadena segura almacenada en el Almacén de parámetros. Tal y como se especifica, CloudFormation usará la versión **10** del parámetro ***IAMUserPassword*** para las operaciones de pila y de conjunto de cambios.

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}}'
```

Obtención de un secreto o un valor secreto de Secrets Manager

Secrets Manager es un servicio que permite almacenar y administrar de forma segura un secreto, como credenciales de base de datos, contraseñas y claves de API de terceros. Con Secrets Manager, puede almacenar y controlar el acceso a estos secretos de forma centralizada, de modo que pueda reemplazar las credenciales codificadas en su código (incluidas las contraseñas), con una llamada API a Secrets Manager para recuperar el secreto mediante programación. Para obtener más información, consulte [¿Qué es AWS Secrets Manager?](#) en la Guía del usuario de AWS Secrets Manager.

Para utilizar secretos completos o valores secretos que están almacenados en Secrets Manager en las plantillas de CloudFormation, debe utilizar referencias dinámicas `secretsmanager`.

Prácticas recomendadas

Siga estas prácticas recomendadas cuando use las referencias dinámicas de Secrets Manager en las plantillas de CloudFormation:

- Utilice referencias sin versión para sus plantillas de CloudFormation: almacene las credenciales en Secrets Manager y utilice referencias dinámicas sin los parámetros `version-stage` o `version-id` para respaldar los flujos de trabajo de rotación de secretos adecuados.
- Aproveche la rotación automática: utilice la característica de rotación automática de Secrets Manager con referencias dinámicas sin versiones para la administración de credenciales. Esto

garantiza que sus credenciales se actualicen periódicamente sin necesidad de cambiar la plantilla. Para obtener más información, consulte [Rotar secretos de AWS Secrets Manager](#).

- Utilice las referencias versionadas con moderación: especifique únicamente parámetros `version-stage` o `version-id` explícitos para situaciones específicas, como situaciones de prueba o reversión.

Consideraciones

Al utilizar referencias dinámicas de `secretsmanager`, hay algunas consideraciones importantes que se deben tener en cuenta:

- CloudFormation no rastrea qué versión de un secreto se usó en implementaciones anteriores. Planifique cuidadosamente su estrategia de administración de secretos antes de implementar referencias dinámicas. Use referencias sin versión siempre que sea posible para aprovechar la rotación automática de secretos. Supervise y valide las actualizaciones de los recursos al realizar cambios en las configuraciones de las referencias dinámicas, por ejemplo, al pasar de referencias dinámicas no versionadas a referencias dinámicas versionadas, y viceversa.
- Cuando se actualiza únicamente el valor secreto en Secrets Manager, CloudFormation recupera automáticamente el nuevo valor. CloudFormation recupera el valor secreto solo durante la creación del recurso o las actualizaciones que modifican el recurso que contiene la referencia dinámica.

Por ejemplo, supongamos que la plantilla incluye un recurso [AWS::RDS::DBInstance](#) en el que la propiedad `MasterPassword` está configurada como una referencia dinámica de Secrets Manager. Tras crear una pila a partir de esta plantilla, se actualiza el valor del secreto en Secrets Manager. Sin embargo, la propiedad `MasterPassword` conserva el valor de la contraseña anterior.

Para aplicar el nuevo valor secreto, tendrá que modificar el recurso `AWS::RDS::DBInstance` en la plantilla de CloudFormation y realizar una actualización de la pila.

Considere la posibilidad de utilizar Secrets Manager para que cambie automáticamente el secreto.

- Las referencias dinámicas de valores seguros, como, por ejemplo, `secretsmanager`, no son compatibles actualmente con los recursos personalizados.
- La referencia dinámica `secretsmanager` se pueden utilizar en todas las propiedades de recursos. El uso de la referencia dinámica `secretsmanager` indica que ni Secrets Manager ni CloudFormation registren o mantengan valores de secretos resueltos. Sin embargo, el valor del

secreto puede mostrarse en el servicio donde se está usando el recurso. Debe revisar el uso para evitar fugas de datos secretos.

Permisos

Para especificar un secreto almacenado en Secrets Manager, debe tener permiso para llamar a [GetSecretValue](#) para el secreto.

Patrón de referencia

Para referenciar secretos de Secrets Manager en una plantilla de CloudFormation, utilice el siguiente patrón de referencia `secretsmanager`.

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

`secret-id`

El nombre o el ARN del secreto.

Para obtener acceso a un secreto de su Cuenta de AWS, solo debe especificar el nombre del secreto. Para obtener acceso a un secreto de otra Cuenta de AWS, especifique el ARN completo del secreto.

Obligatorio.

`secret-string`

El único valor admitido es `SecretString`. El valor predeterminado es `SecretString`.

`json-key`

El nombre de la clave del par clave-valor cuyo valor desea recuperar. Si no se especifica una `json-key`, CloudFormation recupera todo el texto secreto.

Este segmento no puede incluir el signo de dos puntos (`:`).

`version-stage`

La etiqueta de fase de la versión del secreto que se usará. Secrets Manager utiliza etiquetas provisionales para realizar un seguimiento de las diferentes versiones durante el proceso de rotación. Si usa `version-stage`, no especifique `version-id`. Si no especifica `version-stage` ni `version-id`, la versión predeterminada es la `AWSCURRENT`.

Este segmento no puede incluir el signo de dos puntos (:).

version-id

El identificador único de la versión del secreto a utilizar. Si especifica `version-id`, no especifique `version-stage`. Si no especifica `version-stage` ni `version-id`, la versión predeterminada es la `AWSCURRENT`.

Este segmento no puede incluir el signo de dos puntos (:).

Ejemplos

Temas

- [Recuperación de valores de nombre de usuario y contraseña de un secreto](#)
- [Recuperación de la totalidad de SecretString](#)
- [Recuperación de un valor de una versión específica de un secreto](#)
- [Recuperación de secretos de otra Cuenta de AWS](#)

Recuperación de valores de nombre de usuario y contraseña de un secreto

El siguiente ejemplo de [AWS::RDS::DBInstance](#) recupera los valores de nombre de usuario y contraseña almacenados en el secreto *MySecret*. En este ejemplo se muestra el patrón recomendado para las referencias dinámicas sin versión, que utiliza automáticamente la versión `AWSCURRENT` y admite los flujos de trabajo de rotación de Secrets Manager sin necesidad de cambiar la plantilla.

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}}'
```

Recuperación de la totalidad de SecretString

La siguiente referencia dinámica recupera el valor de SecretString para *MySecret*.

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

Otra opción:

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

Recuperación de un valor de una versión específica de un secreto

La siguiente referencia dinámica recupera el valor de *password* para la versión *AWSPREVIOUS* de *MySecret*.

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

Recuperación de secretos de otra Cuenta de AWS

La siguiente referencia dinámica recupera el valor de SecretString para *MySecret* que hay en otra Cuenta de AWS. Debe especificar el ARN del secreto completo para obtener acceso a otra Cuenta de AWS.

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

La siguiente referencia dinámica recupera el valor de *password* para *MySecret* que hay en otra Cuenta de AWS. Debe especificar el ARN del secreto completo para obtener acceso a otra Cuenta de AWS.

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

Obtención de valores de AWS mediante pseudoparámetros

Los pseudoparámetros son variables integradas que proporcionan acceso a información importante del entorno de AWS, como los ID de las cuentas, los nombres de las regiones y los detalles de las pilas, que pueden cambiar de una implementación a otra o de un entorno a otro.

Puede usar pseudoparámetros en lugar de valores codificados de forma rígida para que sus plantillas sean más portátiles y fáciles de reutilizar en diferentes regiones y Cuentas de AWS.

Sintaxis

Puede hacer referencia a los pseudoparámetros mediante la función intrínseca `Ref` o la función intrínseca `Fn::Sub`.

Ref.

La función intrínseca `Ref` utiliza la siguiente sintaxis general. Para obtener más información, consulte [Ref](#).

JSON

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

YAML

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

Fn::Sub

La función intrínseca `Fn::Sub` utiliza un formato diferente que incluye la sintaxis `${}` alrededor del pseudoparámetro. Para obtener más información, consulte [Fn::Sub](#).

JSON

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

YAML

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

Pseudoparámetros disponibles

AWS::AccountId

Devuelve el ID de Cuenta de AWS de la cuenta en la que se está creando la pila como, por ejemplo 123456789012.

Este pseudoparámetro se suele utilizar al definir los roles de IAM, las políticas y otras políticas de recursos que implican ARN específicos de la cuenta.

AWS::NotificationARNs

Devuelve la lista de Nombres de recursos de Amazon (ARN) de los temas de Amazon SNS que reciben notificaciones de eventos de pilas. Puede especificar estos ARN mediante la opción `--notification-arns` en la AWS CLI o a través de la consola mientras crea o actualiza la pila.

A diferencia de otros pseudoparámetros que devuelven un único valor, `AWS::NotificationARNs` devuelve una lista de ARN. Para acceder a un ARN específico de la lista, utilice la función intrínseca `Fn::Select`. Para obtener más información, consulte [Fn::Select](#).

AWS::NoValue

Elimina la propiedad de recurso correspondiente cuando se especifica como valor devuelto en la función intrínseca `Fn::If`. Para obtener más información, consulte [Fn::If](#).

Este pseudoparámetro es particularmente útil para crear propiedades de recursos condicionales que solo deberían incluirse en determinadas condiciones.

AWS::Partition

Devuelve la partición en la que se encuentra el recurso. Para las Regiones de AWS estándar, la partición es `aws`. Para recursos en otras particiones, la partición es `aws-partitionname`. Por

ejemplo, la partición de los recursos de la región China (Pekín y Ningxia) es `aws-cn` y la partición de los recursos en la región AWS GovCloud (Oeste de EE. UU.) es `aws-us-gov`.

La partición forma parte del ARN de los recursos. El uso de `AWS::Partition` garantiza que las plantillas funcionen correctamente en diferentes particiones de AWS.

AWS::Region

Devuelve una cadena que representa la región en la que se crea el recurso que abarca, como por ejemplo `us-west-2`.

Este es uno de los seudoparámetros más utilizados, ya que permite que las plantillas se adapten a diferentes Regiones de AWS sin modificaciones.

AWS::StackId

Devuelve el ID (ARN) de la pila, como `arn:aws:cloudformation:us-west-2:123456789012:stack/teststack/51af3dc0-da77-11e4-872e-1234567db123`.

AWS::StackName

Devuelve el nombre de la pila, como `teststack`.

El nombre de la pila se suele utilizar para crear nombres de recursos únicos que se puedan identificar fácilmente como pertenecientes a una pila específica.

AWS::URLSuffix

Devuelve el sufijo del dominio AWS en la Región de AWS donde se implementa la pila. El sufijo normalmente es `amazonaws.com`, pero para la región China (Pekín), el sufijo es `amazonaws.com.cn`.

Este parámetro resulta especialmente útil a la hora de crear direcciones URL para los puntos de conexión de servicios de AWS.

Ejemplos

Temas

- [Uso básico](#)
- [Uso de AWS::NotificationARNs](#)

- [Propiedades condicionales con AWS::NoValue](#)

Uso básico

En los siguientes ejemplos se crean dos recursos: un tema de Amazon SNS y una alarma de CloudWatch que envía notificaciones a ese tema. Se utiliza `AWS::StackName`, `AWS::Region` y `AWS::AccountId` para insertar dinámicamente el nombre de la pila, la Región de AWS actual y el ID de la cuenta en los nombres, las descripciones y los ARN de los recursos.

JSON

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

YAML

```
Resources:
```

```

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

```

Uso de AWS::NotificationARNs

En los siguientes ejemplos se configura un grupo de escalado automático para enviar notificaciones de eventos de lanzamiento de instancias y errores de lanzamiento. La configuración usa el pseudoparámetro `AWS::NotificationARNs`, que proporciona una lista de los ARN de los temas de Amazon SNS que se especificaron al crear la pila. La función `Fn::Select` elige el primer ARN de esa lista.

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

```

Propiedades condicionales con AWS::NoValue

En el siguiente ejemplo se crea una instancia de base de datos de Amazon RDS que utiliza una instantánea solo si se proporciona un ID de instantánea. Si la condición `UseDBSnapshot` se evalúa como `true`, CloudFormation utiliza el valor del parámetro `DBSnapshotName` para la propiedad `DBSnapshotIdentifier`. Si se evalúa la condición como `false`, CloudFormation elimina la propiedad `DBSnapshotIdentifier`.

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

```

Obtención de salidas exportadas de una pila implementada de CloudFormation

Si tiene varias pilas en la misma Cuenta de AWS y región, es posible que desee compartir información entre ellas. Esto resulta útil cuando una pila necesita usar recursos creados por otra pila.

Por ejemplo, puede tener una pila que cree recursos de red, como subredes y grupos de seguridad, para sus servidores web. Luego, otras pilas que crean los servidores web reales pueden utilizar los recursos de red creados por la primera pila. No es necesario codificar de forma rígida los IDs de recursos en la plantilla de la pila ni pasar los IDs como parámetros de entrada.

Para compartir información entre pilas, exporta los valores de salida de una pila y los importa a otra pila. Así es como funciona:

1. En la plantilla de la primera pila (p. ej., la pila de redes), se definen determinados valores para la exportación mediante el campo `Export` de la sección `Outputs`. Para obtener más información, consulte [Sintaxis de Outputs de la plantilla de CloudFormation](#).
2. Al crear o actualizar esa pila, CloudFormation exporta los valores de salida y los pone a disposición de otras pilas de la misma Cuenta de AWS y región.
3. En la plantilla de la otra pila, se utiliza la función `Fn::ImportValue` para importar los valores exportados de la primera pila.
4. Al crear o actualizar la segunda pila (p. ej., la pila del servidor web), CloudFormation recupera automáticamente los valores exportados de la primera pila y los utiliza.

Para obtener un tutorial y plantillas de ejemplo, consulte [Consulta de las salidas de recursos en otra pila de CloudFormation](#).

Exportación de valores de salida de pilas en comparación con el uso de pilas anidadas

Una pila anidada es una pila que crea en otra pila con el recurso `AWS::CloudFormation::Stack`. Con pilas anidadas, se implementan y administran todos los recursos desde una sola pila. Se pueden usar salidas desde una pila en el grupo de pilas anidadas como entradas a otra pila del grupo. Esto difiere de la exportación de valores.

Si desea que solo se intercambie información dentro de un grupo de pilas anidadas, le sugerimos que utilice pilas anidadas. Para compartir información con otras pilas (no solo dentro del grupo

de pilas anidadas), exporte valores. Por ejemplo, puede crear una sola pila con una subred y, a continuación, exportar su ID. Otras pilas pueden utilizar dicha subred importando su ID. Cada pila no tiene por qué crear su propia subred. Mientras las pilas importen el ID de la subred, no se puede cambiar ni eliminar.

Para obtener más información sobre las pilas anidadas, consulte [División de una plantilla en piezas reutilizables utilizando pilas anidadas](#).

Consideraciones

Las siguientes restricciones se aplican a las referencias cruzadas entre pilas:

- Para cada Cuenta de AWS, los nombres `Export` deben ser únicos dentro de una región.
- No puede crear referencias cruzadas entre regiones. Puede utilizar la función intrínseca `Fn::ImportValue` para importar solo valores que se hayan exportado dentro de la misma región.
- Para las salidas, el valor de la propiedad `Name` de un `Export` no puede usar las funciones `Ref` o `GetAtt` que dependen de un recurso.

Del mismo modo, la función `ImportValue` no puede incluir funciones `Ref` o `GetAtt` que dependen de un recurso.

- Después de que otra pila importe un valor de salida, no se puede eliminar la pila que exporta el valor de salida ni modificar el valor de salida exportado. Todas las importaciones deben eliminarse antes de poder eliminar la pila de exportación o modificar el valor de salida.

Listado de valores de salida exportados

Si necesita ver los valores de las salidas exportadas de sus pilas, utilice uno de los siguientes métodos:

Para obtener una lista de valores de salida exportados (consola)

1. Abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation>.
2. En la barra de navegación de la parte superior de la pantalla, elija la Región de AWS.
3. En el panel de navegación izquierdo, seleccione Exportaciones.

Para obtener una lista de los valores de salida exportados (AWS CLI)

Use el siguiente comando [list-exports](#). Sustituya `us-east-1` por su Región de AWS.

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

A continuación, se muestra un ejemplo del resultado.

```
{
  "Exports": [
    {
      "ExportingStackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/private-vpc/99764070-b56c-xmpl-bee8-062a88d1d800",
      "Name": "private-vpc-subnet-a",
      "Value": "subnet-07b410xmplddcfa03"
    },
    {
      "ExportingStackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/private-vpc/99764070-b56c-xmpl-bee8-062a88d1d800",
      "Name": "private-vpc-subnet-b",
      "Value": "subnet-075ed3xmplebd2fb1"
    },
    {
      "ExportingStackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/private-vpc/99764070-b56c-xmpl-bee8-062a88d1d800",
      "Name": "private-vpc-vpcid",
      "Value": "vpc-011d7xmpl100e9841"
    }
  ]
}
```

CloudFormation muestra los nombres y los valores de las salidas exportadas para la región actual y la pila desde la que se esperaban. Para usar un valor de salida exportado en otra plantilla de pila, puede referenciarla con el nombre de exportación y la función `Fn::ImportValue`.

Enumeración de pilas que importan un valor de salida exportado

Para eliminar o cambiar los valores de salida exportados, primero debe averiguar qué pilas los importan.

Para ver las pilas que importan un valor de salida exportado, utilice uno de los siguientes métodos:

Enumeración de pilas que importan un valor de salida exportado (consola)

1. Abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation>.

2. En el panel de navegación izquierdo, seleccione Exportaciones.
3. Para ver qué pilas importan un valor de exportación determinado, elija Export Name (Nombre de la exportación) para ese valor de exportación. CloudFormation muestra la página de detalles de exportación, que enumera todas las pilas que están importando el valor.

Para obtener una lista de pilas que importan un valor de salida exportado (AWS CLI)

Utilice el comando [list-imports](#). Sustituya *us-east-1* por su Región de AWS y *private-vpc-vpcid* por el nombre del valor de salida exportado.

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

CloudFormation devuelve una lista de pilas que están importando el valor.

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

Una vez que sepa qué pilas importan un valor exportado en particular, debe modificar esas pilas para eliminar las funciones `Fn::ImportValue` que refieren a los valores de salida. Debe eliminar todas las importaciones que hacen referencia a valores de salida exportados antes de poder eliminar o modificar los valores de salida exportados.

Definición de recursos existentes en tiempo de ejecución con tipos de parámetros proporcionados por CloudFormation

Al crear la plantilla, puede crear parámetros que requieran que los usuarios introduzcan identificadores de los recursos de AWS existentes o parámetros de Systems Manager mediante tipos de parámetros especializados proporcionados por CloudFormation.

Temas

- [Descripción general](#)
- [Ejemplo](#)
- [Consideraciones](#)

- [Tipos de parámetros específicos de AWS compatibles](#)
- [Tipos de parámetros de Systems Manager admitidos](#)
- [Tipos de parámetros de Systems Manager no admitidos](#)

Descripción general

En CloudFormation, puede usar parámetros para personalizar las pilas al proporcionar valores de entrada durante la creación o actualización de la pila. Esta característica hace que sus plantillas sean reutilizables y flexibles en diferentes escenarios.

Los parámetros se definen en la sección `Parameters` de una plantilla de CloudFormation. Cada parámetro tiene un nombre y un tipo, y se puede tener varias configuraciones adicionales, como un valor predeterminado y valores permitidos. Para obtener más información, consulte [Sintaxis de Parameters de la plantilla de CloudFormation](#).

El tipo de parámetro determina el tipo de valor de entrada que el parámetro puede aceptar. Por ejemplo, `Number` solo acepta valores numéricos, mientras que `String` acepta la entrada de texto.

CloudFormation proporciona varios tipos de parámetros adicionales que puede usar en sus plantillas para hacer referencia a los recursos de AWS existentes y a los parámetros de Systems Manager.

Estos tipos de parámetro se dividen en dos categorías:

- tipos de parámetros específicos de AWS: CloudFormation proporciona un conjunto de tipos de parámetros que ayudan a atrapar valores no válidos al crear o actualizar una pila. Al utilizar estos tipos de parámetros, cualquier persona que utilice la plantilla debe especificar valores válidos de la Cuenta de AWS y la región en la que van a crear la pila.

Si utilizan la Consola de administración de AWS, CloudFormation proporciona una lista rellena previamente de los valores existentes de su cuenta y región. De esta forma el usuario no tiene que recordar y escribir correctamente un nombre específico o ID. En su lugar, simplemente seleccionan valores de una lista desplegable. En algunos casos, pueden incluso buscar valores por ID, nombre o el valor `Name` de la etiqueta.

- Tipos de parámetros de Systems Manager: CloudFormation también proporciona tipos de parámetros que corresponden a parámetros existentes en el Almacén de parámetros de Systems Manager. Cuando se usan estos tipos de parámetros, cualquier persona que use la plantilla debe especificar una clave de Almacén de parámetros como el valor del tipo de parámetro de Systems Manager y, a continuación, CloudFormation recupera el valor más reciente del Almacén

de parámetros que se usará para su pila. Esto puede resultar útil cuando necesite actualizar con frecuencia las aplicaciones con nuevos valores de propiedad, como los ID de Imagen de máquina de Amazon (AMI). Para obtener más información acerca del Almacén de parámetros, consulte [Almacén de parámetros de Systems Manager](#).

Una vez definidos los parámetros en la sección `Parameters`, puede hacer referencia a los valores de los parámetros en toda la plantilla de CloudFormation mediante la función `Ref`.

Ejemplo

El siguiente ejemplo muestra una plantilla que usa los siguientes tipos de parámetros.

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

Para crear una pila a partir de esta plantilla, debe especificar un ID de VPC, un ID de subred y un nombre de par de claves existentes en su cuenta. También puede especificar una clave del Almacén de parámetros existente que haga referencia al ID de AMI deseado o mantener el valor predeterminado de `/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2`. Este parámetro público es un alias del ID de AMI regional de la AMI más reciente de Amazon Linux 2. Para obtener más información acerca de parámetros públicos, consulte [Descubrir parámetros públicos en el Almacén de parámetros](#) en la Guía del usuario de AWS Systems Manager.

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
```

Comando AWS CLI para crear una pila

El siguiente comando [create-stack](#) crea una pila basada en la plantilla de ejemplo.

```
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"
```

Para utilizar un tipo de parámetro que acepte una lista de cadenas, por ejemplo

List<AWS::EC2::Subnet::Id>, debe evitar las comas del interior del ParameterValue con una doble barra invertida, como se muestra en el siguiente ejemplo.

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

Consideraciones

Recomendamos que use referencias dinámicas para restringir el acceso a las definiciones de configuración confidenciales, como las credenciales de terceros. Para obtener más información, consulte [Obtención de valores almacenados en otros servicios con referencias dinámicas](#).

Si quiere permitir a los usuarios de la plantilla especificar valores de distintas Cuentas de AWS, no utilice tipos de parámetros específicos de AWS. En su lugar, defina los parámetros de tipo String o CommaDelimitedList.

Hay algunas cosas que se deben tener en cuenta con los tipos de parámetros de Systems Manager:

- Puede ver los valores resueltos de los parámetros [describe-stacks](#) en la pestaña Parámetros de la pila de la consola o mediante la ejecución de [describe-change-set](#). Tenga en cuenta que estos valores se establecen al crearse o actualizarse la pila, por lo que pueden diferir de los valores más recientes del Almacén de parámetros.

- En el caso de las actualizaciones de pilas, si se utiliza la opción Usar un valor existente (o se establece UsePreviousValue como verdadero), significa que se quiere seguir utilizando la misma clave del Almacén de parámetros, no su valor. CloudFormation siempre recupera el valor más reciente.
- Si especifica cualquier valor permitido u otras restricciones, CloudFormation los valida con las claves de parámetros que especifique, pero no con sus valores. Debe validar los valores en el propio Almacén de parámetros.
- Al crear o actualizar pilas y crear conjuntos de cambios, CloudFormation usa los valores que existan en el Almacén de parámetros en ese momento. Si un parámetro especificado no existe en el Almacén de parámetros con la Cuenta de AWS del intermediario, CloudFormation devuelve un error de validación.
- Al ejecutar un conjunto de cambios, CloudFormation usa los valores que se especifican en el conjunto de cambios. Debe revisar estos valores antes de ejecutar el conjunto de cambios porque pueden cambiar en el almacén de parámetros entre el momento de la creación del conjunto de cambios y su ejecución.
- Para los parámetros almacenados en el mismo Cuenta de AWS, debe proporcionar el nombre del parámetro. Para los parámetros del almacén de parámetros compartidos por otra Cuenta de AWS, debe proporcionar el ARN completo del parámetro.

Tipos de parámetros específicos de AWS compatibles

CloudFormation es compatible con los siguientes tipos específicos de AWS:

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

Una zona de disponibilidad, como por ejemplo `us-west-2a`.

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

Un ID de imagen de Amazon EC2, como por ejemplo `ami-0ff8a91507f77f867`. Tenga en cuenta que la consola de CloudFormation no muestra una lista desplegable de valores para este tipo de parámetro.

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

Un ID de instancia Amazon EC2, como por ejemplo `i-1e731a32`.

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

Un nombre del par de claves de Amazon EC2.

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

Un nombre del grupo de seguridad de VPC predeterminado; por ejemplo, `my-sg-abc`.

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

Un ID de grupo de seguridad, como por ejemplo `sg-a123fd85`.

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

Un ID de subred, como por ejemplo `subnet-123a351e`.

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

Un ID de volumen de Amazon EBS, como por ejemplo `vol-3cdd3f56`.

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

Un ID de VPC, como por ejemplo `vpc-a123baa3`.

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

Un ID de zona alojada de Amazon Route 53, como por ejemplo `Z23YXV40VPL04A`.

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

Una matriz de las zonas de disponibilidad de una región, como por ejemplo `us-west-2a`, `us-west-2b`.

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

Una matriz de ID de imagen de Amazon EC2, como por ejemplo `ami-0ff8a91507f77f867`, `ami-0a584ac55a7631c0c`. Tenga en cuenta que la consola de CloudFormation no muestra una lista desplegable de valores para este tipo de parámetro.

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

Una matriz de ID de instancia Amazon EC2, como por ejemplo `i-1e731a32`, `i-1e731a34`.

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

Una matriz de los nombres de los grupos de seguridad de VPC predeterminados; por ejemplo, `my-sg-abc`, `my-sg-def`.

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

Una matriz de los ID de grupos de seguridad, como por ejemplo `sg-a123fd85`, `sg-b456fd85`.

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

Una matriz de los IDs de subred, como por ejemplo subnet-123a351e, subnet-456b351e.

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

Una matriz de ID de volumen de Amazon EBS, como por ejemplo vol-3cdd3f56, vol-4cdd3f56.

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

Una matriz de los ID de VPC, como por ejemplo vpc-a123baa3, vpc-b456baa3.

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

Una matriz de ID de zonas alojadas de Amazon Route 53, como por ejemplo Z23YXV40VPL04A, Z23YXV40VPL04B.

Tipos de parámetros de Systems Manager admitidos

CloudFormation admite los siguientes tipos de parámetros de Systems Manager:

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

El nombre de una clave de parámetro de Systems Manager. Utilice este tipo de parámetro únicamente para comprobar que existe un parámetro obligatorio. CloudFormation no recuperará el valor real asociado con el parámetro.

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

Un parámetro de Systems Manager cuyo valor es una cadena. Esto corresponde al tipo de parámetro String en Parameter Store.

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

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

Un parámetro de Administrador de sistemas cuyo valor es una lista de cadenas. Esto corresponde al tipo de parámetro StringList en Parameter Store.

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

Un parámetro de Systems Manager cuyo valor es un tipo de parámetro específico de AWS.

Por ejemplo, lo siguiente especifica el tipo `AWS::EC2::KeyPair::KeyName`:

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

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

Un parámetro de Systems Manager cuyo valor es una lista de tipos de parámetros específicos de AWS.

Por ejemplo, lo siguiente especifica una lista de tipos `AWS::EC2::KeyPair::KeyName`:

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

Tipos de parámetros de Systems Manager no admitidos

CloudFormation no admite los siguientes tipos de parámetros de Systems Manager:

- Listas de tipos de parámetros de Systems Manager; por ejemplo:

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

Además, CloudFormation no es compatible con la definición de parámetros de plantillas como tipos de parámetros de Systems Manager `SecureString`. Sin embargo, también puede especificar cadenas seguras como valores de parámetros para algunos recursos. Para obtener más información, consulte [Obtención de valores almacenados en otros servicios con referencias dinámicas](#).

Tutoriales de CloudFormation

Esta documentación proporciona una colección de tutoriales diseñados para que pueda practicar de forma práctica con las implementaciones de pilas.

- [Consulta de las salidas de recursos en otra pila de CloudFormation](#): en este tutorial, se muestra cómo hacer referencia a los resultados de una pila de CloudFormation dentro de otra pila. En lugar de incluir todos los recursos en una sola pila, puede crear recursos de AWS en pilas independientes para crear plantillas más modulares y reutilizables.
- [Implementación de aplicaciones en Amazon EC2](#): obtenga información sobre cómo usar CloudFormation para instalar, configurar e iniciar automáticamente su aplicación en instancias de Amazon EC2. De esta manera, puede duplicar fácilmente implementaciones y actualizar instalaciones existentes sin conectarse directamente a la instancia.
- [Actualización de una pila de CloudFormation](#): consulte una serie sencilla de actualizaciones a una pila en ejecución con CloudFormation.

- [Creación de una aplicación con escalado y equilibrio de carga](#): descubra cómo usar CloudFormation para crear una aplicación escalable y con equilibrio de carga. En este tutorial se explica cómo crear un grupo de escalado automático, un equilibrador de carga y otros recursos relacionados para garantizar que la aplicación pueda gestionar cargas de tráfico variables y mantener una alta disponibilidad.
- [Emparejamiento con una VPC en otra Cuenta de AWS](#): este tutorial le guía a través del proceso de creación de una conexión de emparejamiento de nube privada virtual (VPC) entre dos VPC en diferentes Cuentas de AWS. El emparejamiento de VPC le ayuda a dirigir el tráfico entre las VPC y acceder a los recursos como si formaran parte de la misma red.
- [Implementaciones azul/verde de ECS a través de CodeDeploy con CloudFormation](#): descubra cómo usar CloudFormation para realizar implementaciones azul/verde AWS CodeDeploy en Amazon ECS. Las implementaciones azul/verde son una forma de actualizar sus aplicaciones o servicios con un tiempo de inactividad mínimo.

Consulta de las salidas de recursos en otra pila de CloudFormation

En este tutorial, se muestra cómo hacer referencia a los resultados de una pila de CloudFormation dentro de otra pila para crear plantillas más modulares y reutilizables.

En lugar de incluir todos los recursos en una sola pila, debe crear recursos de AWS en pilas independientes. A continuación, puede consultar salidas de recursos necesarios de otras pilas. Al restringir referencias de pila cruzadas a salidas, usted controla las partes de una pila referenciadas por otras pilas.

Por ejemplo, podría tener una pila de red con una VPC, un grupo de seguridad, y una subred para aplicaciones web públicas y una pila de aplicaciones web públicas independientes. Para garantizar que las aplicaciones web utilicen el grupo de seguridad y la subred de la pila de la red, debe crear una referencia de pila cruzada que permita a la pila de aplicaciones web hacer referencia a resultados de recursos desde la pila de red. Con una referencia de pila cruzada, los propietarios de las pilas de aplicaciones web no necesitan crear ni mantener reglas de red o activos.

Para crear una referencia de pila cruzada, utilice el campo de campo de salida `Export` para marcar el valor de la salida de un recurso para exportar. A continuación, utilice la función intrínseca `Fn::ImportValue` para importar el valor. Para obtener más información, consulte [Obtención de salidas exportadas de una pila implementada de CloudFormation](#).

 Note

CloudFormation es un servicio gratuito. Sin embargo, se le cobrará por los recursos de AWS que incluya en sus pilas a la tarifa actual para cada una. Para obtener más información sobre los precios de AWS, consulte [la página de detalles de cada producto](#).

Temas

- [Uso de una plantilla de ejemplo para crear una pila de red](#)
- [Uso de una plantilla de ejemplo para crear una pila de aplicaciones web](#)
- [Verificación de que la pila funcione según lo esperado](#)
- [Solución de problemas de asignación de AMI](#)
- [Eliminación de sus recursos](#)

Uso de una plantilla de ejemplo para crear una pila de red

Antes de empezar este tutorial, compruebe que tiene permisos de IAM para utilizar todos los siguientes servicios: Amazon VPC, Amazon EC2 y CloudFormation.

La pila de red contiene la VPC, el grupo de seguridad y la subred que utilizará en la pila de aplicaciones web. Además de estos recursos, la pila de red crea una gateway de Internet y las tablas de ruteo para permitir el acceso público.

Debe crear esta pila antes de crear la pila de aplicaciones web. Si crea la pila de aplicaciones web primero, no tendrá un grupo de seguridad ni una subred.

La plantilla de pila está disponible a través de la siguiente URL: <https://s3.amazonaws.com/cloudformation-examples/user-guide/cross-stack/SampleNetworkCrossStack.template>. Para ver los recursos que creará la pila, elija el enlace, que abre la plantilla. En la sección Outputs, puede ver los recursos de red que la plantilla de ejemplo exporta. Los nombres de los recursos exportados están prefijados con el nombre de la pila en caso de que se exporten recursos de red de otras pilas. Cuando los usuarios importan recursos de red, pueden especificar de qué pilas se importan los recursos.

Para crear la pila de la red

1. Inicie sesión en la Consola de administración de AWS y abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation>.

2. En la página Pilas, elija Crear pila en la parte superior derecha y, a continuación, elija Con recursos nuevos (estándar).
3. Elija Seleccione una plantilla existente y, en la sección Especificar plantilla, seleccione Amazon S3 URL.
4. En Amazon S3 URL, pegue la siguiente URL: **`https://s3.amazonaws.com/cloudformation-examples/user-guide/cross-stack/SampleNetworkCrossStack.template`**.
5. Elija Siguiente.
6. En Stack name (Nombre de pila), escriba **SampleNetworkCrossStack** y, a continuación, elija Next (Siguiente).

 Note

Registre el nombre de esta pila. Necesitará el nombre de la pila al lanzar la pila de aplicaciones web.

7. Elija Siguiente. Para este tutorial, no tiene que añadir etiquetas ni especificar una configuración avanzada.
8. Asegúrese de que el nombre de la pila y la URL de la plantilla sean correctos y elija Create stack (Crear pila).

CloudFormation puede tardar varios minutos en crear la pila. Espere hasta que todos los recursos se hayan creado correctamente antes de proceder a crear la pila de aplicaciones web.

9. Para monitorizar el progreso, vea los eventos de la pila. Para obtener más información, consulte [Monitorizar el progreso de la pila](#).

Uso de una plantilla de ejemplo para crear una pila de aplicaciones web

La pila de aplicaciones web crea una instancia de EC2 que utiliza el grupo de seguridad y la subred desde la pila de la red.

Debe crear esta pila en la misma Región de AWS que la pila de la red.

La plantilla de pila está disponible a través de la siguiente URL: <https://s3.amazonaws.com/cloudformation-examples/user-guide/cross-stack/SampleWebAppCrossStack.template>. Para ver los recursos que creará la pila, elija el enlace, que abrirá la plantilla. En la sección Resources, vea las

propiedades de la instancia de EC2. Puede ver cómo se importan los recursos de red de otra pila usando la función `Fn::ImportValue`.

Para crear la pila de aplicaciones web

1. En la página Pilas, seleccione Crear pila en la parte superior derecha y, a continuación, seleccione Con recursos nuevos (estándar).
2. Elija Seleccione una plantilla existente y, en la sección Especificar plantilla, seleccione Amazon S3 URL.
3. En Amazon S3 URL, pegue la siguiente URL: **`https://s3.amazonaws.com/cloudformation-examples/user-guide/cross-stack/SampleWebAppCrossStack.template`**.
4. Elija Siguiente.
5. Para Stack name (Nombre de pila), escriba **`SampleWebAppCrossStack`**. En la sección Parameters (Parámetros), utilice el valor predeterminado para el parámetro `NetworkStackName` y, a continuación, elija Next (Siguiente).

La plantilla de ejemplo utiliza el valor del parámetro para especificar en cada pila para importar valores.

6. Elija Siguiente. Para este tutorial, no tiene que añadir etiquetas ni especificar una configuración avanzada.
7. Asegúrese de que el nombre de la pila y la URL de la plantilla sean correctos y elija Create stack (Crear pila).

CloudFormation puede tardar varios minutos en crear la pila.

Verificación de que la pila funcione según lo esperado

Después de haber creado la pila, vea sus recursos y tenga en cuenta el ID de instancia. Para obtener más información acerca de la visualización de recursos de la pila, consulte [Visualización de la información de la pila desde la consola de CloudFormation](#).

Para verificar la subred y el grupo de seguridad de la instancia, vea las propiedades de la instancia en la [consola de Amazon EC2](#). Si la instancia utiliza el grupo de seguridad y la subred en la pila `SampleNetworkCrossStack`, se ha creado correctamente una referencia de pila cruzada.

Utilice la consola para ver los resultados de pila y la URL del sitio web de ejemplo para comprobar que se ejecuta la aplicación web. Para obtener más información, consulte [Visualización de la información de la pila desde la consola de CloudFormation](#).

Solución de problemas de asignación de AMI

Si recibe el error `Template error: Unable to get mapping for AWSRegionArch2AMI::[region]::HVM64`, la plantilla no incluye ninguna asignación de AMI para su Región de AWS. En lugar de actualizar la asignación, recomendamos utilizar los parámetros públicos de Systems Manager para hacer referencia a las AMI más recientes de forma dinámica:

1. Descargue la plantilla `SampleWebAppCrossStack` en su máquina local desde <https://s3.amazonaws.com/cloudformation-examples/user-guide/cross-stack/SampleWebAppCrossStack.template>.
2. Elimine toda la sección de asignación de `AWSRegionArch2AMI`.
3. Agregue el siguiente parámetro de Systems Manager:

```
"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. Reemplace la referencia a `ImageId` existente:

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

Por:

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

Este parámetro se resuelve automáticamente en la AMI de Amazon Linux 2 más reciente para la región donde se implementa la pila.

Para otras distribuciones de Linux, utilice la ruta de parámetros apropiada. Para obtener más información, consulte [Descubrir parámetros públicos en Parameter Store](#) en la Guía del usuario de AWS Systems Manager.

5. Cargue la plantilla modificada en un bucket de Amazon S3 de su cuenta:

```
aws s3 cp SampleWebAppCrossStack.template s3://amzn-s3-demo-bucket/
```

6. Al crear la pila, especifique la URL de la plantilla de S3 en lugar de la URL de ejemplo.

Eliminación de sus recursos

Para asegurarse de que no se le cobra por servicios no deseados, elimine las pilas.

Para eliminar las pilas

1. En la consola de CloudFormation, seleccione la pila `SampleWebAppCrossStack`.
2. Elija Actions (Acciones) y, a continuación, elija Delete Stack (Eliminar pila).
3. En el mensaje de confirmación, elija Delete (Eliminar).
4. Una vez que se haya eliminado la pila, repita los mismos pasos con la pila `SampleNetworkCrossStack`.

Note

Espere hasta que CloudFormation elimine por completo la pila `SampleWebAppCrossStack`. Si la instancia de EC2 sigue ejecutándose en la VPC, CloudFormation no eliminará la VPC en la pila `SampleNetworkCrossStack`.

Implementación de aplicaciones en Amazon EC2

Puede usar CloudFormation para instalar, configurar e iniciar automáticamente aplicaciones en instancias de Amazon EC2. Esto le permite duplicar fácilmente implementaciones y actualizar instalaciones existentes sin conectarse directamente a la instancia, lo que permite ahorrar mucho tiempo y esfuerzo.

CloudFormation incluye un conjunto de scripts auxiliares (`cfn-init`, `cfn-signal`, `cfn-get-metadata` y `cfn-hup`) que se basan en `cloud-init`. Se llama a estos scripts auxiliares desde las plantillas de CloudFormation para instalar, configurar y actualizar aplicaciones en instancias Amazon EC2 que se encuentran en la misma plantilla. Para obtener más información, consulte [Referencia de scripts auxiliares de CloudFormation](#) en la Guía de referencia de plantillas de CloudFormation.

En el [tutorial de introducción](#), creó un servidor web sencillo utilizando UserData con un script de bash básico. Si bien esto funcionó para una página sencilla de “Hello World”, las aplicaciones reales suelen necesitar una configuración más sofisticada, que incluya:

- Varios paquetes de software instalados en el orden correcto.
- Archivos de configuración complejos creados con contenido específico.
- Servicios iniciados y configurados para iniciarse automáticamente.
- Gestión de errores y validación del proceso de configuración.

Los scripts auxiliares de CloudFormation proporcionan una forma más sólida y sostenible de configurar las instancias de EC2 en comparación con los scripts de bash básicos en UserData. El script auxiliar `cfn-init` lee los datos de configuración de los metadatos de la plantilla y los aplica sistemáticamente a la instancia.

En este tutorial, aprenderá a utilizar el script auxiliar `cfn-init` y a supervisar el proceso de arranque.

Note

CloudFormation es gratuito, pero se le cobrarán los recursos de Amazon EC2 que cree. Sin embargo, si es la primera vez que utiliza AWS, puede aprovechar el [nivel gratuito](#) para minimizar o eliminar los costos durante este proceso de aprendizaje.

Temas

- [Requisitos previos](#)
- [Descripción de los conceptos de arranque](#)
- [Inicio con un ejemplo sencillo de arranque](#)
- [Cómo agregar archivos y comandos](#)
- [Cómo agregar la seguridad a la red](#)
- [La plantilla de arranque completa](#)
- [Creación de la pila con la consola](#)
- [Supervisión del proceso de arranque](#)
- [Prueba del servidor web arrancado](#)
- [Solución de problemas de arranque](#)

- [Eliminar recursos](#)
- [Siguiendo pasos](#)

Requisitos previos

- Debe haber completado el tutorial [Creación de su primera pila](#) o una experiencia equivalente con los conceptos básicos de CloudFormation.
- Debe tener acceso a una Cuenta de AWS con un usuario o rol de IAM que tenga permisos para usar Amazon EC2 y CloudFormation, o acceso de usuario administrativo.
- Debe tener una nube privada virtual (VPC) que tenga acceso a Internet. Esta plantilla de tutorial requiere una VPC predeterminada, que se incluye automáticamente con las Cuentas de AWS más recientes. Si no tiene una VPC predeterminada o si se ha eliminado, consulte la sección de solución de problemas en el tutorial [Creación de su primera pila](#) para ver soluciones alternativas.

Descripción de los conceptos de arranque

Vamos a describir los conceptos clave que hacen que el arranque funcione antes de crear la plantilla.

El script auxiliar **cfn-init**

CloudFormation proporciona scripts auxiliares de Python que puede utilizar para instalar software e iniciar servicios en una instancia de Amazon EC2. El script `cfn-init` lee los metadatos de los recursos de su plantilla y aplica la configuración a su instancia.

El proceso funciona de la siguiente manera:

1. Define la configuración en la sección Metadata de su recurso de EC2.
2. Llama a `cfn-init` desde el script `UserData`.
3. `cfn-init` lee los metadatos y aplica la configuración.
4. La instancia se configura de acuerdo con sus especificaciones.

Estructura de los metadatos

La configuración se define en una estructura específica dentro de su instancia de EC2.

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

El script `cfn-init` procesa estas secciones en un orden específico: paquetes, grupos, usuarios, orígenes, archivos, comandos y, a continuación, servicios.

Inicio con un ejemplo sencillo de arranque

Comencemos con un ejemplo mínimo de arranque que simplemente instale e inicie Apache.

```

Resources:
  EC2Instance:
    Type: AWS::EC2::Instance
    Metadata:
      AWS::CloudFormation::Init:
        config:
          packages:
            # Install Apache web server
          yum:
            httpd: []
          services:
            # Start Apache and enable it to start on boot
            systemd:
              httpd:
                enabled: true
                ensureRunning: true
    Properties:
      ImageId: !Ref LatestAmiId
      InstanceType: !Ref InstanceType
      UserData: !Base64
        # Script that runs when instance starts
        Fn::Sub: |
          #!/bin/bash
          yum install -y aws-cfn-bootstrap
          /opt/aws/bin/cfn-init -v --stack ${AWS::StackName} --resource EC2Instance --
          region ${AWS::Region}

```

En este ejemplo sencillo se ilustran los conceptos básicos:

- La sección `packages` instala el paquete `httpd` usando `yum`. Esto funciona en Amazon Linux y en otras distribuciones de Linux que utilizan `yum`.
- La sección `services` garantiza que `httpd` se inicie y ejecute automáticamente.
- `UserData` instala las herramientas de arranque más recientes y llama a `cfn-init`.

Cómo agregar archivos y comandos

Ahora, para mejorar nuestro ejemplo, vamos a agregar una página web personalizada y un archivo de registro en el directorio `/var/log` de la instancia de EC2.

Creación de archivos

La sección `files` le permite crear archivos en la instancia con contenido específico. La barra vertical (`|`) le permite pasar un bloque literal de texto (código HTML) como contenido del archivo (`/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>
```

Ejecución de comandos

La sección `commands` le permite ejecutar comandos del intérprete de comandos durante el proceso de arranque. Este comando crea un archivo de registro en `/var/log/welcome.txt` en la instancia de EC2. Para verlo, necesita un par de claves de Amazon EC2 que utilizará para el acceso mediante SSH y un rango de direcciones IP que se pueda utilizar para acceder mediante SSH a la instancia (no se trata aquí).

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

Cómo agregar la seguridad a la red

Como vamos a configurar un servidor web, debemos permitir que el tráfico web (HTTP) llegue a nuestra instancia de EC2. Para ello, crearemos un grupo de seguridad que permita el tráfico

entrante en el puerto 80 desde su dirección IP. Las instancias de EC2 también tienen que enviar tráfico a Internet, por ejemplo, para instalar actualizaciones de paquetes. De forma predeterminada, los grupos de seguridad permiten el tráfico saliente. A continuación, asociaremos este grupo de seguridad a nuestra instancia de EC2 mediante la propiedad `SecurityGroupIds`.

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

La plantilla de arranque completa

Ahora vamos a juntar todas las partes. Esta es la plantilla completa en la que se combinan todos los conceptos que hemos tratado.

```
AWS::TemplateFormatVersion: 2010-09-09
Description: Bootstrap an EC2 instance with Apache web server using cfn-init

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

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

  MyIP:
    Description: Your IP address in CIDR format (e.g. 203.0.113.1/32)
    Type: String
```

```

MinLength: 9
MaxLength: 18
Default: 0.0.0.0/0
AllowedPattern: '^(\d{1,3}\.){3}\d{1,3}\.\/\d{1,2}$'
ConstraintDescription: must be a valid IP CIDR range of the form x.x.x.x/x.

```

Resources:

WebServerSecurityGroup:

```
Type: AWS::EC2::SecurityGroup
```

Properties:

```
GroupDescription: Allow HTTP access from my IP address
```

SecurityGroupIngress:

```

- IpProtocol: tcp
  Description: HTTP
  FromPort: 80
  ToPort: 80
  CidrIp: !Ref MyIP

```

WebServer:

```
Type: AWS::EC2::Instance
```

Metadata:

```
AWS::CloudFormation::Init:
```

config:

packages:

yum:

```
httpd: []
```

files:

```
/var/www/html/index.html:
```

```
content: |
```

```
<body>
```

```
<h1>Congratulations, you have successfully launched the AWS
```

```
CloudFormation sample.</h1>
```

```
</body>
```

commands:

createWelcomeLog:

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

services:

systemd:

httpd:

```
enabled: true
```

```
ensureRunning: true
```

Properties:

```
ImageId: !Ref LatestAmiId
```

```
InstanceType: !Ref InstanceType
```

```
SecurityGroupIds:
  - !Ref WebServerSecurityGroup
UserData: !Base64
  Fn::Sub: |
    #!/bin/bash
    yum install -y aws-cfn-bootstrap
    /opt/aws/bin/cfn-init -v --stack ${AWS::StackName} --resource WebServer --
region ${AWS::Region}
Tags:
  - Key: Name
    Value: Bootstrap Tutorial Web Server

Outputs:
  WebsiteURL:
    Value: !Sub 'http://${WebServer.PublicDnsName}'
    Description: EC2 instance public DNS name
```

Creación de la pila con la consola

El siguiente procedimiento implica cargar la plantilla de pila de ejemplo desde un archivo. Abra un editor de texto en su equipo local y agregue la plantilla. Guarde el archivo con el nombre `samplelinux2stack.template`.

Para lanzar la plantilla de pila

1. Inicie sesión en la Consola de administración de AWS y abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation>.
2. Elija Create stack (Crear pila), With new resources (standard) (Con nuevos recursos [estándar]).
3. En Especificar plantilla, seleccione Cargar un archivo de plantilla y, a continuación, seleccione Elegir archivo para cargar el archivo `samplelinux2stack.template`.
4. Elija Siguiente.
5. En la página Especificar los detalles de la pila, ingrese **BootstrapTutorialStack** como nombre de la pila.
6. En Parámetros, haga lo siguiente.
 - LatestAmild: deje el valor predeterminado.
 - InstanceType: elija t2.micro o t3.micro para el tipo de instancia de EC2.
 - MyIP: ingrese su dirección IP pública real con el sufijo /32.
7. Elija Siguiente dos veces y, luego, Enviar para crear la pila.

Supervisión del proceso de arranque

Los procesos de arranque tardan más que los lanzamientos sencillos de EC2 porque se instala y configura software adicional.

Supervisión del progreso del arranque

1. En la consola de CloudFormation, seleccione su pila y abra la pestaña Eventos.
2. Preste atención al evento `WebServer CREATE_IN_PROGRESS`. El proceso de arranque comienza después del lanzamiento de la instancia.
3. El proceso de arranque normalmente dura unos minutos. Verá `WebServer CREATE_COMPLETE` cuando haya terminado.

Si quiere ver lo que ocurre durante el proceso de arranque, puede consultar los registros de la instancia.

Visualización de los registros de arranque (opcional)

1. Abra la [consola de EC2](#) y busque su instancia.
2. Seleccione la instancia y, luego, elija Acciones, Monitoreo y solución de problemas, Obtener registro del sistema para ver el progreso del arranque.
3. Si no ve los registros inmediatamente, espere y actualice la página.

Prueba del servidor web arrancado

Cuando la pila muestre `CREATE_COMPLETE`, pruebe su servidor web.

Prueba del servidor web

1. En la consola de CloudFormation, vaya a la pestaña Salidas de su pila.
2. Haga clic en el valor de `WebsiteURL` para abrir su servidor web en una pestaña nueva.
3. Debería ver la página web personalizada con el mensaje `Congratulations, you have successfully launched the AWS CloudFormation sample`.

 Note

Si la página no se carga inmediatamente, espere un minuto e inténtelo de nuevo. Es posible que el proceso de arranque aún se esté completando incluso después de que la pila muestre `CREATE_COMPLETE`.

Solución de problemas de arranque

Si el proceso de arranque falla o el servidor web no funciona, aquí tiene algunos problemas y soluciones comunes.

Problemas comunes

- No se puede crear la pila: consulte la pestaña Eventos para ver mensajes de error específicos.
- No se puede acceder al servidor web: compruebe que la dirección IP sea correcta en el parámetro `MyIP`. Recuerde incluir `/32` al final.
- No se puede completar el proceso de arranque: es posible que la instancia se lance, pero `cf-init` falla. Compruebe los registros del sistema tal como se describe en la sección de supervisión.

Eliminar recursos

Para evitar cargos continuos, puede llevar a cabo una limpieza eliminando la pila y sus recursos.

Eliminación de la pila y sus recursos

1. Abra la [consola de CloudFormation](#).
2. En la página Pilas, seleccione la opción situada junto al nombre de la pila que ha creado (**BootstrapTutorialStack**) y, a continuación, seleccione Eliminar.
3. Cuando se le pida confirmación, seleccione Eliminar.
4. Supervise el progreso del proceso de eliminación de la pila en la pestaña Evento. El estado de **BootstrapTutorialStack** cambia a `DELETE_IN_PROGRESS`. Cuando CloudFormation completa la eliminación de la pila, quita la pila de la lista.

Siguientes pasos

¡Enhorabuena! Ha aprendido correctamente a arrancar instancias de EC2 con CloudFormation. Ahora sabe:

- Cómo usar los scripts auxiliares de `cfn-init`.
- Cómo estructurar los metadatos para el arranque.
- Cómo instalar paquetes, crear archivos, ejecutar comandos y administrar servicios.
- Cómo supervisar si se producen problemas de arranque.

Para seguir aprendiendo:

- Aprenda a actualizar una pila en ejecución y a usar el script auxiliar `cfn-hup`. Para obtener más información, consulte [Actualización de una pila de CloudFormation](#).
- Aprenda a arrancar una pila de Windows. Para obtener más información, consulte [Arranque de pilas de CloudFormation basadas en Windows](#).
- Explore escenarios de arranque más complejos con varios conjuntos de configuración. Para obtener más información, consulte [cfn-init](#) y [AWS::CloudFormation::Init](#) en la Guía de referencia de plantillas de CloudFormation.
- Obtenga información sobre cómo usar `cfn-signal` para informar sobre el estado de finalización del arranque. Para obtener más información, consulte [cfn-signal](#) en la Guía de referencia de plantillas de CloudFormation.

Actualización de una pila de CloudFormation

Note

Este tutorial se basa en los conceptos presentados en el tutorial [Implementación de aplicaciones en Amazon EC2](#). Si no ha completado ese tutorial, le recomendamos que lo haga primero para comprender el arranque de EC2 con CloudFormation.

En este tema se ilustra una sencilla progresión de actualizaciones a una pila en ejecución. Le explicaremos con detalle los pasos siguientes:

1. Cree la pila inicial: cree una pila con una AMI base de Amazon Linux 2 e instale el Apache Web Server y una aplicación PHP sencilla con los scripts auxiliares de CloudFormation.
2. Actualice la aplicación: actualice uno de los archivos en la aplicación e implemente el software con CloudFormation.
3. Añada un par de claves: añada un par de claves de Amazon EC2 a la instancia y, a continuación, actualice el grupo de seguridad para permitir el acceso de SSH a la instancia.
4. Actualice el tipo de instancia: cambie el tipo de instancia de la instancia de Amazon EC2 subyacente.
5. Actualice la AMI: cambie la Imagen de máquina de Amazon (AMI) para la instancia de Amazon EC2 en la pila.

Note

CloudFormation es gratuito, pero se le cobrarán los recursos de Amazon EC2 que cree. Sin embargo, si es la primera vez que utiliza AWS, puede aprovechar el [nivel gratuito](#) para minimizar o eliminar los costos durante este proceso de aprendizaje.

Temas

- [Paso 1: crear la pila inicial](#)
- [Paso 2: actualizar la aplicación](#)
- [Paso 3: añadir acceso SSH con un par de claves](#)
- [Paso 4: actualizar el tipo de instancia](#)
- [Paso 5: actualizar la AMI](#)
- [Consideraciones sobre el impacto y la disponibilidad](#)
- [Recursos relacionados](#)

Paso 1: crear la pila inicial

Empezaremos creando una pila que podemos utilizar a lo largo del resto de este tema. Hemos proporcionado una sencilla plantilla que lanza una aplicación web PHP de una sola instancia alojada en el Apache Web Server y que se ejecuta en una AMI de Amazon Linux 2.

El Apache Web Server, PHP y la aplicación PHP sencilla están instalados por los scripts auxiliares de CloudFormation que vienen instalados de forma predeterminada en la AMI de Amazon Linux 2. El

siguiente fragmento de la plantilla muestra los metadatos que describe los paquetes y archivos para instalar, en este caso, el Apache Web Server y la infraestructura PHP desde el repositorio Yum para la AMI de Amazon Linux 2. El fragmento también muestra la sección `Services`, que garantiza que el Apache Web Server se está ejecutando.

```
WebServerInstance:
  Type: AWS::EC2::Instance
  Metadata:
    AWS::CloudFormation::Init:
      config:
        packages:
          yum:
            httpd: []
            php: []
        files:
          /var/www/html/index.php:
            content: |
              <?php
                echo '<h1>Hello World!</h1>';
              ?>
            mode: '000644'
            owner: apache
            group: apache
        services:
          systemd:
            httpd:
              enabled: true
              ensureRunning: true
```

La aplicación en sí es un ejemplo del tipo “Hola, mundo”, que se define completamente en la plantilla. En el caso de una aplicación real, los archivos pueden almacenarse en Amazon S3, GitHub o en otro repositorio y consultarse desde la plantilla. CloudFormation puede descargar paquetes (como RPM o RubyGems) y los archivos individuales de referencia y expandir archivos `.zip` y `.tar` para crear los artefactos de aplicación en la instancia de Amazon EC2.

La plantilla permite y configura el daemon `cfn-hup` para escuchar los cambios de la configuración definidos en los metadatos de la instancia de Amazon EC2. Al utilizar el daemon `cfn-hup`, puede actualizar el software de la aplicación, como la versión de Apache o PHP, o actualizar el archivo de la aplicación PHP directamente desde CloudFormation. El siguiente fragmento del mismo recurso de Amazon EC2 de la plantilla muestra los elementos necesarios para configurar `cfn-hup` para que

llame a `cfn-init` cada dos minutos para detectar y aplicar actualizaciones a los metadatos. De lo contrario, `cfn-init` solo se ejecuta una vez al iniciarse.

```
files:
  /etc/cfn/cfn-hup.conf:
    content: !Sub |
      [main]
      stack=${AWS::StackId}
      region=${AWS::Region}
      # The interval used to check for changes to the resource metadata in minutes.
      Default is 15
      interval=2
      mode: '000400'
      owner: root
      group: root
  /etc/cfn/hooks.d/cfn-auto-reloader.conf:
    content: !Sub |
      [cfn-auto-reloader-hook]
      triggers=post.update
      path=Resources.WebServerInstance.Metadata.AWS::CloudFormation::Init
      action=/opt/aws/bin/cfn-init -s ${AWS::StackId} -r WebServerInstance --region
      ${AWS::Region}
      runas=root
services:
  systemd:
    cfn-hup:
      enabled: true
      ensureRunning: true
      files:
        - /etc/cfn/cfn-hup.conf
        - /etc/cfn/hooks.d/cfn-auto-reloader.conf
```

Para completar la pila, en la sección **Properties** de la definición de la instancia de Amazon EC2, la propiedad `UserData` contiene el script `cloud-init` que llama a `cfn-init` para instalar los paquetes y archivos. Para obtener más información, consulte [Referencia de scripts auxiliares de CloudFormation](#) en la Guía de referencia de plantillas de CloudFormation. La plantilla también crea un grupo de seguridad de Amazon EC2

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

```
Parameters:
```

```
  LatestAmiId:
```

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

InstanceType:

Description: WebServer EC2 instance type

Type: String

Default: t3.micro

AllowedValues:

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

ConstraintDescription: must be a valid EC2 instance type.

Resources:**WebServerInstance:**

Type: AWS::EC2::Instance

Properties:

ImageId: !Ref LatestAmiId

InstanceType: !Ref InstanceType

SecurityGroupIds:

- !Ref WebServerSecurityGroup

UserData:

Fn::Base64: !Sub |

```

#!/bin/bash -xe
# Get the latest CloudFormation package
yum update -y aws-cfn-bootstrap
# Run cfn-init
/opt/aws/bin/cfn-init -v --stack ${AWS::StackName} --resource
WebServerInstance --region ${AWS::Region} || error_exit 'Failed to run cfn-init'

# Start up the cfn-hup daemon to listen for changes to the EC2 instance
metadata
/opt/aws/bin/cfn-hup || error_exit 'Failed to start cfn-hup'
# Signal success or failure
/opt/aws/bin/cfn-signal -e $? --stack ${AWS::StackName} --resource
WebServerInstance --region ${AWS::Region}
Metadata:
  AWS::CloudFormation::Init:
    config:
      packages:
        yum:
          httpd: []
          php: []
      files:
        /var/www/html/index.php:
          content: |
            <?php
            echo "<h1>Hello World!</h1>";
            ?>
          mode: '000644'
          owner: apache
          group: apache
        /etc/cfn/cfn-hup.conf:
          content: !Sub |
            [main]
            stack=${AWS::StackId}
            region=${AWS::Region}
            # The interval used to check for changes to the resource metadata in
minutes. Default is 15
            interval=2
          mode: '000400'
          owner: root
          group: root
        /etc/cfn/hooks.d/cfn-auto-reloader.conf:
          content: !Sub |
            [cfn-auto-reloader-hook]
            triggers=post.update

```

```

        path=Resources.WebServerInstance.Metadata.AWS::CloudFormation::Init
        action=/opt/aws/bin/cfn-init -s ${AWS::StackId} -r WebServerInstance --
region ${AWS::Region}
        runas=root
    services:
        systemd:
            httpd:
                enabled: true
                ensureRunning: true
            cfn-hup:
                enabled: true
                ensureRunning: true
            files:
                - /etc/cfn/cfn-hup.conf
                - /etc/cfn/hooks.d/cfn-auto-reloader.conf
    CreationPolicy:
        ResourceSignal:
        Timeout: PT5M

WebServerSecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
        GroupDescription: Enable HTTP access via port 80
        SecurityGroupIngress:
            - IpProtocol: tcp
              FromPort: 80
              ToPort: 80
              CidrIp: 0.0.0.0/0

Outputs:
    WebsiteURL:
        Value: !Sub 'http://${WebServerInstance.PublicDnsName}'
        Description: URL of the web application

```

Lanzamiento de una pila a partir de esta plantilla

1. Copie la plantilla y guárdela localmente en el sistema como un archivo de texto. Anote la ubicación, porque necesitará usar el archivo en el siguiente paso.
2. Inicie sesión en la Consola de administración de AWS y abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation>.
3. Elija Crear pila, Con nuevos recursos (estándar).
4. Elija Elegir una plantilla existente.

5. En Especificar plantilla, elija Cargar un archivo de plantilla, busque el archivo que ha creado en el primer paso y, a continuación, seleccione Siguiente.
6. En la página Especificar detalles de pila, escriba **UpdateTutorial** como nombre de la pila.
7. En Parámetros, mantenga todos los parámetros iguales y elija Siguiente dos veces.
8. En la pantalla Revisar y añadir, elija Enviar.

Cuando el estado de la pila sea CREATE_COMPLETE, la pestaña Salidas mostrará la URL de su sitio web. Si selecciona el valor de salida de WebsiteURL, verá su nueva aplicación PHP en marcha.

Paso 2: actualizar la aplicación

Ahora que hemos implementado la pila, vamos a actualizar la aplicación. Haremos un sencillo cambio en el texto impreso por la aplicación. Para ello, agregaremos un comando echo para el archivo index.php tal y como se muestra en este fragmento de la plantilla:

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

Utilice un editor de texto para modificar manualmente el archivo de plantilla que guardó localmente.

Ahora, actualice la pila.

Actualización de la pila con la plantilla actualizada

1. En la consola de CloudFormation, seleccione la pila de **UpdateTutorial**.
2. Seleccione Actualizar y realizar una actualización directa.
3. Seleccione Sustituir plantilla existente.
4. En el panel Especificar plantilla, elija Cargar un archivo de plantilla, cargue el archivo de plantilla modificado y, a continuación, elija Siguiente.

5. En la página Especificar detalles de pila, modifique los parámetros y, a continuación, elija Siguiente.
6. En la página Revisar, revise los cambios. En Cambios, debería ver a CloudFormation actualizar el recurso `WebServerInstance`.
7. Elija Enviar.

Cuando la pila está en el estado `UPDATE_COMPLETE`, puede seleccionar el valor de salida de `WebsiteURL` de nuevo, para verificar si han entrado en vigor los cambios de la aplicación. El daemon `cfn-hup` se ejecuta cada 2 minutos, por lo que la aplicación puede tardar hasta 2 minutos en cambiar una vez actualizado la pila.

Para ver el conjunto de recursos que se actualizaron, vaya a la consola de CloudFormation. En la pestaña Events (Eventos), examine los eventos de pila. En este caso particular, se actualizaron los metadatos de la instancia `WebServerInstance` de Amazon EC2, lo que provocó que CloudFormation también reevaluara los demás recursos (`WebServerSecurityGroup`) para asegurarse de que no hubiera otros cambios. Ninguno de los otros recursos de la pila fue modificado. CloudFormation solo actualizará aquellos recursos en la pila que se vean afectados por cualquier cambio realizado en la pila. Estos cambios pueden ser directos, como los cambios de metadatos o la propiedad, o pueden deberse a dependencias o flujos de datos a través de `Ref`, `GetAtt` u otras funciones de plantilla intrínsecas. Para obtener más información, consulte la [Referencia de función intrínseca](#).

Esta sencilla actualización ilustra el proceso. Sin embargo, puede realizar cambios mucho más complejos en los archivos y paquetes implementados en las instancias de Amazon EC2. Por ejemplo, puede decidir que necesita añadir MySQL a la instancia, junto con el soporte de PHP para MySQL. Para hacerlo, solo tiene que añadir los archivos y paquetes adicionales junto con los servicios adicionales a la configuración y, luego, actualizar la pila para implementar los cambios.

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

    ...
```

```
services:
  systemd:
    httpd:
      enabled: true
      ensureRunning: true
    cfn-hup:
      enabled: true
      ensureRunning: true
      files:
        - /etc/cfn/cfn-hup.conf
        - /etc/cfn/hooks.d/cfn-auto-reloader.conf
  mysqld:
    enabled: true
    ensureRunning: true
```

Puede actualizar los metadatos de CloudFormation de modo que se actualicen según las nuevas versiones de los paquetes que utiliza la aplicación. En los ejemplos anteriores, la propiedad de la versión de cada paquete está vacía, lo que indica que `cfn-init` debería instalar la última versión del paquete.

```
packages:
  yum:
    httpd: []
    php: []
```

Si lo desea, puede especificar una cadena de versión de un paquete. Si cambia la cadena de versión en las posteriores llamadas de actualización de pila, se implementará la nueva versión del paquete. A continuación se muestra un ejemplo de cómo utilizar los números de versión para paquetes RubyGems. Cualquier paquete compatible con control de versiones puede tener versiones específicas.

```
packages:
  rubygems:
    mysql: []
    rubygems-update:
      - "1.6.2"
    rake:
      - "0.8.7"
    rails:
      - "2.3.11"
```

Paso 3: añadir acceso SSH con un par de claves

También puede actualizar un recurso de la plantilla para añadir propiedades que no estaban especificadas originalmente en la plantilla. Para ilustrarlo, vamos a añadir un par de claves de Amazon EC2 a una instancia de EC2 existente y, a continuación, abrir el puerto 22 en el grupo de seguridad de Amazon EC2, para que pueda utilizar Secure Shell (SSH) para acceder a la instancia.

Para añadir el acceso SSH a una instancia Amazon EC2 existente

1. Añada dos parámetros adicionales a la plantilla para transmitirlo en el nombre de una ubicación SSH y un par de claves Amazon EC2 existentes.

```
Parameters:
  KeyName:
    Description: Name of an existing EC2 KeyPair to enable SSH access to the
instance
    Type: AWS::EC2::KeyPair::KeyName
    ConstraintDescription: must be the name of an existing EC2 KeyPair.

  SSHLocation:
    Description: The IP address that can be used to SSH to the EC2 instances in
CIDR format (e.g. 203.0.113.1/32)
    Type: String
    MinLength: 9
    MaxLength: 18
    Default: 0.0.0.0/0
    AllowedPattern: '^(\d{1,3}\.){3}\d{1,3}\.\d{1,2}$'
    ConstraintDescription: must be a valid IP CIDR range of the form x.x.x.x/x.
```

2. Añada la propiedad KeyName a la instancia de Amazon EC2.

```
WebServerInstance:
  Type: AWS::EC2::Instance
  Properties:
    ImageId: !Ref LatestAmiId
    InstanceType: !Ref InstanceType
    KeyName: !Ref KeyName
    SecurityGroupIds:
      - !Ref WebServerSecurityGroup
```

3. Añada el puerto 22 y la ubicación de SSH a las reglas de entrada del grupo de seguridad de Amazon EC2.


```
WebServerSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Enable HTTP access via port 80 and SSH access via port 22
    SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: 80
        ToPort: 80
        CidrIp: 0.0.0.0/0
      - IpProtocol: tcp
        FromPort: 22
        ToPort: 22
        CidrIp: !Ref SSHLocation
```

4. Para actualizar la pila, siga los mismos pasos que se explican en [Paso 2: actualizar la aplicación](#).

Paso 4: actualizar el tipo de instancia

Ahora vamos a demostrar cómo actualizar la infraestructura subyacente cambiando el tipo de instancia.

La pila que hemos creado utiliza hasta ahora una instancia de Amazon EC2 t3.micro. Supongamos que el sitio web recién creado obtiene más tráfico de lo que puede controlar una instancia t3.micro y ahora desea pasar a un tipo de instancia de Amazon EC2 m5.large. Si la arquitectura del tipo de instancia cambia, la instancia se debe crear con una AMI diferente. Sin embargo, tanto el t3.micro como el m5.large utilizan las mismas arquitecturas de CPU y ejecutan las AMI de Amazon Linux 2 (x86_64). Para obtener más información, consulte [Compatibilidad para cambiar el tipo de instancias](#) en la Guía del usuario de Amazon EC2.

Vamos a utilizar la plantilla que hemos modificado en el paso anterior para cambiar el tipo de instancia. Dado que InstanceType fue un parámetro de entrada de la plantilla, no es necesario modificar la plantilla; se puede cambiar el valor del parámetro en la página Especificar parámetros.

Actualización de la pila con un nuevo valor de parámetro

1. En la consola de CloudFormation, seleccione la pila de **UpdateTutorial**.
2. Seleccione Actualizar y realizar una actualización directa.
3. Seleccione Usar plantilla actual y, a continuación, elija Siguiente.

4. En la página Especificar los detalles de la pila, cambie el valor del cuadro de texto InstanceType de t3.micro a m5.large. A continuación, elija Siguiente dos veces.
5. En la página Revisar, revise los cambios. En Cambios, debería ver a CloudFormation actualizar el recurso WebServerInstance.
6. Elija Enviar.

Puede cambiar de manera dinámica el tipo de instancia de una instancia de Amazon EC2 con respaldo de EBS al iniciar y detener la instancia. CloudFormation intenta optimizar el cambio al actualizar el tipo de instancia y reiniciar la instancia, de modo que el ID de la instancia no cambie. Cuando la instancia se reinicia, sin embargo, la dirección IP pública de la instancia sí cambia. Para garantizar que la dirección IP elástica se vincule correctamente después del cambio, CloudFormation también actualizará la dirección IP elástica. Puede ver los cambios en la consola de CloudFormation en la pestaña de eventos.

Para comprobar el tipo de instancia desde Consola de administración de AWS, abra la consola de Amazon EC2 y busque allí su instancia.

Paso 5: actualizar la AMI

Ahora, actualicemos nuestra pila para usar Amazon Linux 2023, que es la próxima generación de Amazon Linux.

La actualización de la AMI es un cambio importante que requiere reemplazar la instancia. No es posible iniciar y detener la instancia para modificar la AMI. CloudFormation lo considera un cambio en una propiedad inmutable del recurso. Para realizar un cambio en una propiedad inmutable, CloudFormation debe lanzar un recurso de reemplazo, en este caso, una nueva instancia de Amazon EC2 que ejecute la nueva AMI.

Veamos cómo podemos actualizar nuestra plantilla de pila para usar Amazon Linux 2023. Los cambios clave incluyen la actualización del parámetro de AMI y el cambio de un administrador de paquetes yum a dnf.

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

```
Parameters:
```

```
  LatestAmiId:
```

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

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

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

InstanceType:

Description: WebServer EC2 instance type

Type: String

Default: t3.micro

AllowedValues:

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

ConstraintDescription: must be a valid EC2 instance type.

KeyName:

Description: Name of an existing EC2 KeyPair to enable SSH access to the instance

Type: AWS::EC2::KeyPair::KeyName

ConstraintDescription: must be the name of an existing EC2 KeyPair.

SSHLocation:

Description: The IP address that can be used to SSH to the EC2 instances in CIDR format (e.g. 203.0.113.1/32)

Type: String

MinLength: 9

MaxLength: 18

Default: 0.0.0.0/0

AllowedPattern: '^(\d{1,3}\.){3}\d{1,3}\s\/\d{1,2}\$'

ConstraintDescription: must be a valid IP CIDR range of the form x.x.x.x/x.

Resources:

WebServerInstance:

Type: AWS::EC2::Instance

Properties:

ImageId: !Ref LatestAmiId

InstanceType: !Ref InstanceType

KeyName: !Ref KeyName

SecurityGroupIds:

- !Ref WebServerSecurityGroup

UserData:

Fn::Base64: !Sub |

#!/bin/bash -xe

Get the latest CloudFormation package

dnf update -y aws-cfn-bootstrap

Run cfn-init

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

WebServerInstance --region \${AWS::Region} || error_exit 'Failed to run cfn-init'

Start up the cfn-hup daemon to listen for changes to the EC2 instance

metadata

/opt/aws/bin/cfn-hup || error_exit 'Failed to start cfn-hup'

Signal success or failure

/opt/aws/bin/cfn-signal -e \$? --stack \${AWS::StackName} --resource

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

Metadata:

AWS::CloudFormation::Init:

config:

packages:

dnf:

httpd: []

php: []

files:

/var/www/html/index.php:

content: |

<?php

echo "<h1>Hello World!</h1>";

echo "<p>This is an updated version of our application.</p>";

echo "<p>Running on Amazon Linux 2023!</p>";

?>

mode: '000644'

owner: apache

group: apache

```

    /etc/cfn/cfn-hup.conf:
      content: !Sub |
        [main]
        stack=${AWS::StackId}
        region=${AWS::Region}
        # The interval used to check for changes to the resource metadata in
minutes. Default is 15
        interval=2
        mode: '000400'
        owner: root
        group: root
    /etc/cfn/hooks.d/cfn-auto-reloader.conf:
      content: !Sub |
        [cfn-auto-reloader-hook]
        triggers=post.update
        path=Resources.WebServerInstance.Metadata.AWS::CloudFormation::Init
        action=/opt/aws/bin/cfn-init -s ${AWS::StackId} -r WebServerInstance --
region ${AWS::Region}
        runas=root
  services:
    systemd:
      httpd:
        enabled: true
        ensureRunning: true
      cfn-hup:
        enabled: true
        ensureRunning: true
      files:
        - /etc/cfn/cfn-hup.conf
        - /etc/cfn/hooks.d/cfn-auto-reloader.conf
  CreationPolicy:
    ResourceSignal:
      Timeout: PT5M

WebServerSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Enable HTTP access via port 80 and SSH access via port 22
    SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: 80
        ToPort: 80
        CidrIp: 0.0.0.0/0
      - IpProtocol: tcp

```

```
FromPort: 22
ToPort: 22
CidrIp: !Ref SSHLocation
```

Outputs:

```
WebsiteURL:
  Value: !Sub 'http://${WebServerInstance.PublicDnsName}'
  Description: URL of the web application
```

Para actualizar la pila, siga los mismos pasos que se explican en [Paso 2: actualizar la aplicación](#).

Una vez que la nueva instancia está en funcionamiento, CloudFormation actualiza los demás recursos de la pila para que se vinculen al nuevo recurso. Cuando se crean todos los nuevos recursos, se elimina el recurso anterior, un proceso conocido como UPDATE_CLEANUP. En este momento, podrá observar que el ID de instancia y el URL de la aplicación de la instancia de la pila han cambiado como resultado de la actualización. Los acontecimientos de la tabla de eventos contienen una descripción “La actualización solicitada tiene un cambio de una propiedad inmutable y, por tanto, se ha creado un nuevo recurso físico” para indicar que se reemplazó un recurso.

Alternativa: si tiene un código de aplicación escrito en la AMI que desea actualizar, puede utilizar el mismo mecanismo de actualización de pila para actualizar la AMI y así cargar su nueva aplicación.

Actualización de la AMI con un código de la aplicación personalizado

1. Cree nuevas AMI que contengan los cambios del sistema operativo o su aplicación. Para obtener más información, consulte [Creación de una AMI basada en Amazon EBS](#) en la Guía del usuario de Amazon EC2.
2. Actualice la plantilla para incorporar los nuevos ID de AMI.
3. Para actualizar la pila, siga los mismos pasos que se explican en [Paso 2: actualizar la aplicación](#).

Al actualizar la pila, CloudFormation detecta que el ID de AMI ha cambiado y, a continuación, se activa una actualización de la pila de la misma forma que iniciamos el anterior.

Consideraciones sobre el impacto y la disponibilidad

Las distintas propiedades afectan de forma diferente a los recursos de la pila. Puede utilizar CloudFormation para actualizar cualquier propiedad; no obstante, antes de realizar cambios, debe tener en cuenta estas preguntas:

1. ¿Cómo afecta la actualización al propio recurso? Por ejemplo, la actualización de un umbral de alarma inutilizará la alarma durante la actualización. Como hemos visto, cambiar el tipo de instancia requiere que la instancia se detenga y se reinicie. CloudFormation utiliza las acciones de actualización o modificación para los recursos subyacentes con el propósito de realizar cambios en los recursos. Para comprender el impacto de las actualizaciones, debería comprobar la documentación de los recursos específicos.
2. ¿Es el cambio mutable o inmutable? Algunos cambios en las propiedades de recursos, como, por ejemplo, el cambio de la AMI en una instancia de Amazon EC2, no son compatibles con los servicios subyacentes. En el caso de cambios mutables, CloudFormation utilizará las API de tipo actualización o modificación para los recursos subyacentes. Para cambios de propiedad inmutables, CloudFormation creará nuevos recursos con las propiedades actualizadas y, a continuación, vincularlas a la pila antes de eliminar los antiguos recursos. Aunque CloudFormation intenta reducir el tiempo de inactividad de los recursos de la pila; sustituir un recurso es un proceso de varios pasos y llevará tiempo. Durante la reconfiguración de pila, su aplicación no estará totalmente operativa. Por ejemplo, es posible que no pueda atender a las solicitudes ni acceder a una base de datos.

Recursos relacionados

Para obtener más información acerca del uso de CloudFormation para iniciar las aplicaciones y de la integración con otros servicios de configuración e implementación como Puppet y Opscode Chef, consulte los siguientes documentos técnicos:

- [Proceso de arranque de aplicaciones mediante CloudFormation](#)
- [Integración de CloudFormation con Opscode Chef](#)
- [Integración de CloudFormation con Puppet](#)

Creación de una aplicación con escalado y equilibrio de carga

Para este tutorial, creará una pila que lo ayudará a configurar una aplicación con escalado y equilibrio de carga. El tutorial proporciona una plantilla de ejemplo que utilizará para crear la pila. La plantilla de ejemplo proporciona un grupo de escalado automático, un equilibrador de carga de aplicación, grupos de seguridad que controlan el tráfico al equilibrador de carga y al grupo de escalado automático y una configuración de notificaciones de Amazon SNS para publicar notificaciones sobre las actividades de escalado.

Esta plantilla crea una o varias instancias de Amazon EC2 y un equilibrador de carga de aplicación. Se le facturarán los recursos de AWS que utilice si crea una pila a partir de esta plantilla.

Plantilla de pila completa

Empecemos por la plantilla.

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"
        }
    }
}
}

```



```
}  
}
```

Tutorial de plantilla

La primera parte de esta plantilla especifica los `Parameters`. A cada parámetro se le debe asignar un valor en tiempo de ejecución para que CloudFormation aprovisiona correctamente la pila. Los recursos especificados más adelante en la plantilla hacen referencia a estos valores y utilizan los datos.

- `InstanceType`: el tipo de instancia de EC2 que Amazon EC2 Auto Scaling aprovisiona. Si no se especifica, se usa el valor predeterminado de `t3.micro`.
- `KeyName`: un par de claves de EC2 existente para permitir el acceso SSH a las instancias.
- `LatestAmiId`: el ID de imagen de máquina de Amazon (AMI) para las instancias. Si no se especifica, las instancias se lanzan con una AMI de Amazon Linux 2, mediante un parámetro AWS Systems Manager público mantenido por AWS. Para obtener más información, consulte [Buscar parámetros públicos](#) en la Guía del usuario de AWS Systems Manager.
- `OperatorEmail`: la dirección de correo electrónico a la que desea enviar notificaciones de actividad de escalado.
- `SSHLocation`: el intervalo de direcciones IP que se puede usar para enviar SSH a las instancias.
- `Subnets`: al menos dos subredes en diferentes zonas de disponibilidad.
- `VPC`: una nube privada virtual (VPC) en su cuenta que permite a los recursos de subredes públicas conectarse a Internet.

Note

Puede usar la VPC predeterminada y las subredes predeterminadas para permitir que las instancias accedan a Internet. Si utiliza su propia VPC, asegúrese de que tiene una subred asignada a cada zona de disponibilidad de la región de en la que esté trabajando. Como mínimo, debe tener dos subredes públicas disponibles para crear el balanceador de carga.

La siguiente parte de esta plantilla especifica los `Resources`. Esta sección especifica los recursos de la pila y sus propiedades.

[AWS::EC2::SecurityGroup](#) de recurso de `ELBSecurityGroup`

- `SecurityGroupIngress` contiene una regla de entrada TCP que permite el acceso desde todas las direcciones IP (`"CidrIp": "0.0.0.0/0"`) en el puerto 80.

[AWS::EC2::SecurityGroup](#) de recurso de `EC2SecurityGroup`

- `SecurityGroupIngress` contiene dos reglas de entrada: 1) una regla de entrada TCP que permite el acceso SSH (puerto 22) desde el rango de direcciones IP que usted proporciona para el parámetro de entrada de `SSHLocation` y 2) una regla de entrada TCP que permite el acceso desde el equilibrador de carga al especificar el grupo de seguridad del equilibrador de carga. La función [GetAtt](#) se usa para obtener el ID del grupo de seguridad con el nombre lógico `ELBSecurityGroup`.

Recurso `EC2TargetGroup` de [AWS::ElasticLoadBalancingV2::TargetGroup](#)

- `Port`, `Protocol` y `HealthCheckProtocol` especifican el puerto de instancia de EC2 (80) y el protocolo (HTTP) a los que el `ApplicationLoadBalancer` enruta el tráfico y que Elastic Load Balancing usa para comprobar el estado de las instancias de EC2.
- `HealthCheckIntervalSeconds` especifica que las instancias de EC2 tengan un intervalo de 30 segundos entre las comprobaciones de estado. `HealthCheckTimeoutSeconds` se define como el periodo de tiempo en el que Elastic Load Balancing espera una respuesta del destino de comprobación de estado (15 segundos en este ejemplo). Cuando termina el tiempo de espera, Elastic Load Balancing marca la comprobación de estado de las instancias de EC2 como en mal estado. Cuando una instancia de EC2 falla tres comprobaciones consecutivas (`UnhealthyThresholdCount`), Elastic Load Balancing detiene el tráfico de enrutamiento a dicha instancia de EC2 hasta que esa instancia tenga cinco comprobaciones correctas consecutivas (`HealthyThresholdCount`). En ese momento, Elastic Load Balancing considera que la instancia tiene un estado correcto y comienza a redirigir el tráfico a la instancia nuevamente.
- `TargetGroupAttributes` actualiza el valor de retraso de anulación del registro del grupo objetivo a 20 segundos. De forma predeterminada, Elastic Load Balancing espera 300 segundos para completar el proceso de anulación del registro.

Recurso `ALBListener` de [AWS::ElasticLoadBalancingV2::Listener](#)

- `DefaultActions` especifica el puerto que el equilibrador de carga escucha, el grupo objetivo al que el equilibrador de carga reenvía las solicitudes y el protocolo utilizado para enrutar solicitudes.

Recurso `ApplicationLoadBalancer` de [AWS::ElasticLoadBalancingV2::LoadBalancer](#)

- `Subnets` toma el valor del parámetro de entrada `Subnets` como la lista de subredes públicas donde se crearán los nodos del equilibrador de carga.
- `SecurityGroup` obtiene el ID del grupo de seguridad que funciona como un firewall virtual para los nodos del equilibrador de carga con el objetivo de controlar el tráfico entrante. La función [GetAtt](#) se usa para obtener el ID del grupo de seguridad con el nombre lógico `ELBSecurityGroup`.

Recurso `LaunchTemplate` de [AWS::EC2::LaunchTemplate](#)

- `ImageId` toma el valor del parámetro de entrada `LatestAmiId` como la AMI que se va a utilizar.
- `KeyName` adopta el valor del parámetro de entrada `KeyName` como el par de claves de EC2 que se usará.
- `SecurityGroupIds` obtiene el ID del grupo de seguridad con el nombre lógico `EC2SecurityGroup` que funciona como un firewall virtual para las instancias de EC2 con el objetivo de controlar el tráfico entrante.
- `UserData` es un script de configuración que se ejecuta una vez que la instancia está operativa y en ejecución. En este ejemplo, el script instala Apache y crea un archivo `index.html`.

Recurso `NotificationTopic` de [AWS::SNS::Topic](#)

- `Subscription` toma el valor del parámetro de entrada `OperatorEmail` como la dirección de correo electrónico del destinatario de las notificaciones cuando hay actividades de escalado.

Recurso `WebServerGroup` de [AWS::AutoScaling::AutoScalingGroup](#)

- `MinSize` y `MaxSize` establecen el número mínimo y máximo de instancias de EC2 en el grupo de escalado automático.
- `TargetGroupARNs` toma el ARN del grupo de destino con el nombre lógico `EC2TargetGroup`. A medida que este grupo de escalado automático escala, registra y anula automáticamente las instancias en este grupo de destino.
- `VPCZoneIdentifier` toma el valor del parámetro de entrada `Subnets` como la lista de subredes públicas donde pueden crearse las instancias de EC2.

Paso 1: Lance la pila

Antes de lanzar la pila, compruebe que tiene permisos de AWS Identity and Access Management (IAM) para utilizar todos los siguientes servicios: Amazon EC2, Amazon EC2 Auto Scaling, AWS Systems Manager, Elastic Load Balancing, Amazon SNS y CloudFormation.

El siguiente procedimiento implica cargar la plantilla de pila de ejemplo desde un archivo. Abra un editor de texto en su equipo local y agregue una de las plantillas. Guarde el archivo con el nombre `sampleloadbalancedappstack.template`.

Para lanzar la plantilla de pila

1. Inicie sesión en la Consola de administración de AWS y abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation>.
2. Elija Create stack (Crear pila), With new resources (standard) (Con nuevos recursos [estándar]).
3. En Especificar plantilla, seleccione Cargar un archivo de plantilla y, a continuación, seleccione Elegir archivo para cargar el archivo `sampleloadbalancedappstack.template`.
4. Elija Siguiente.
5. En la página Especificar detalles de pila, en Nombre de pila, escriba un nombre para la pila (por ejemplo, **SampleLoadBalancedAppStack**).
6. En Parámetros, revise los parámetros de la pila y proporcione valores para todos los parámetros que no tengan valores predeterminados, incluidos OperatorEmail, SSHLocation, KeyName, VPC y Subnets.
7. Seleccione Next (Siguiente) dos veces.
8. En la página Revisar, revise y confirme la configuración.
9. Elija Enviar.

Puede ver el estado de la pila en la consola de CloudFormation en la columna Estado. Cuando CloudFormation haya creado correctamente la pila, recibirá el estado CREATE_COMPLETE.

Note

Después de crear la pila, debe confirmar la suscripción antes de que la dirección de correo electrónico pueda comenzar a recibir notificaciones. Para obtener más información, consulte [Obtener notificaciones de Amazon SNS cuando su grupo de escalado automático escala](#) en la Guía del usuario de Amazon EC2 Auto Scaling.

Paso 2: Eliminar los recursos de ejemplo

Para asegurarse de que no se cobra por recursos de ejemplo no utilizados, elimine la pila.

Para eliminar la pila

1. En la consola de CloudFormation, elija la pila SampleLoadBalancedAppStack.
2. Elija Eliminar.
3. En el mensaje de confirmación, elija Eliminar pila.

El estado de SampleLoadBalancedAppStack cambia a DELETE_IN_PROGRESS. Cuando CloudFormation completa la eliminación de la pila, quita la pila de la lista.

Use la plantilla de ejemplo de este tutorial para crear sus propias plantillas de pila. Para obtener más información, consulte [Tutorial: Configuración de una aplicación con escalado y equilibrio de carga aplicados](#) en la Guía del usuario de Amazon EC2 Auto Scaling.

Emparejamiento con una VPC en otra Cuenta de AWS

Puede emparejar con una nube privada virtual (VPC) en otra Cuenta de AWS mediante el uso de [AWS::EC2::VPCPeeringConnection](#). Esto crea una conexión de red entre dos VPC que permite dirigir el tráfico entre ellas, para que puedan comunicarse como si estuviesen en la misma red. Una interconexión de VPC puede ayudar a facilitar el acceso a los datos y la transferencia de datos.

Para establecer una conexión de emparejamiento de VPC, tendrá que autorizar dos Cuentas de AWS distintas en una sola pila de CloudFormation.

Para obtener más información sobre el emparejamiento de VPC y sus limitaciones, consulte la [Guía de emparejamiento de VPC de Amazon](#).

Requisitos previos

1. Necesita un ID de VPC del mismo nivel, un ID de Cuenta de AWS del mismo nivel y un [rol de acceso entre cuentas](#) para la conexión de emparejamiento.

 Note

Este tutorial hace referencia a dos cuentas: en primer lugar una cuenta que permite interconexiones entre cuentas (la cuenta del aceptador). En segundo lugar una cuenta que solicite la interconexión (la cuenta del solicitante).

2. Para aceptar la interconexión de VPC, la función de acceso entre cuentas debe ser asumible por usted. El recurso se comporta de la misma forma que el recurso de una interconexión de VPC en la misma cuenta. Para obtener más información sobre cómo un administrador de IAM concede permisos para asumir el rol entre cuentas, consulte [Conceder permisos de usuario para cambiar de rol](#) en la Guía del usuario de IAM.

Paso 1: Cree una VPC y un rol entre cuentas

En este paso, creará la VPC y el rol en la cuenta del aceptador.

Crear una VPC y un rol de acceso entre cuentas

1. Inicie sesión en la Consola de administración de AWS y abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation>.
2. En la página Pilas, seleccione Crear pila en la parte superior derecha y, a continuación, seleccione Con recursos nuevos (estándar).
3. En Requisito previo: preparar plantilla, elija Seleccione una plantilla existente y, a continuación, Cargar un archivo de plantilla, Elegir archivo.
4. Abra un editor de texto en su equipo local y agregue una de las siguientes plantillas. Guarde el archivo y vuelva a la consola para seleccionarlo como archivo de plantilla.

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. Elija Siguiente.
6. Asigne a la pila un nombre (por ejemplo, **VPC-owner**) y luego escriba el ID de la Cuenta de AWS de la cuenta del solicitante en el campo PeerRequesterAccountId.
7. Acepte los valores predeterminados y, a continuación, elija Next (Siguiente).
8. Elija Confirmo que CloudFormation puede crear recursos de IAM y, a continuación, elija Crear pila.

Paso 2: creación de una plantilla que incluya **AWS::EC2::VPCPeeringConnection**

Ahora que ha creado la VPC y un rol entre cuentas, puede interconectar con la VPC a través de otra Cuenta de AWS (la cuenta de solicitante).

Crear una plantilla que incluya el recurso [AWS::EC2::VPCPeeringConnection](#)

1. Vuelva a la página de inicio de la consola CloudFormation.
2. En la página Pilas, seleccione Crear pila en la parte superior derecha y, a continuación, seleccione Con recursos nuevos (estándar).
3. En Requisito previo: preparar plantilla, elija Seleccione una plantilla existente y, a continuación, Cargar un archivo de plantilla, Elegir archivo.

4. Abra un editor de texto en su equipo local y agregue una de las siguientes plantillas. Guarde el archivo y vuelva a la consola para seleccionarlo como archivo de plantilla.

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. Elija Siguiente.
6. Dé un nombre a la pila (por ejemplo, **VPC-peering-connection**).
7. Acepte los valores predeterminados y, a continuación, elija Next (Siguiente).
8. Elija Confirmo que CloudFormation puede crear recursos de IAM y, a continuación, elija Crear pila.

Creación de una plantilla con una política muy restrictiva

Es posible que le interese crear una política muy restrictiva para las interconexiones de la VPC con otra Cuenta de AWS.

El siguiente ejemplo de plantilla muestra cómo cambiar la plantilla de propietario de mismo nivel de VPC (la cuenta del aceptador creada en el paso 1 anterior) para que sea más restrictiva.

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/${vpc}'
          - Action: 'ec2:acceptVpcPeeringConnection'
            Condition:
              StringEquals:
                'ec2:AcceptorVpc':
                  'Fn::Sub': 'arn:aws:ec2:${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

```

Para acceder a la VPC, puede utilizar la misma plantilla de solicitante como en el paso 2 anterior.

Para obtener más información, consulte [Administración de identidades y acceso para la interconexión de VPC](#) en la Guía de interconexión de VPC de Amazon.

Implementaciones azul/verde de ECS a través de CodeDeploy con CloudFormation

Para actualizar una aplicación que se ejecuta en Amazon Elastic Container Service (Amazon ECS), puede usar una estrategia de implementación azul/verde de CodeDeploy. Esta estrategia minimiza las interrupciones causadas por los cambios en las versiones de las aplicaciones.

En una implementación azul/verde, se crea un nuevo entorno de aplicaciones (denominado verde) junto con el entorno activo actual (denominado azul). Esto permite supervisar y probar el entorno verde antes de enrutar el tráfico en vivo del entorno azul al entorno verde. Una vez que el entorno verde dé servicio al tráfico en directo, puede terminar el entorno azul de forma segura.

Para realizar implementaciones azul/verde de CodeDeploy en ECS con CloudFormation, incluya la siguiente información en la plantilla de pila:

- Una sección `Hooks` que describe un enlace `AWS::CodeDeploy::BlueGreen`.
- Una sección `Transform` que especifica la transformación `AWS::CodeDeployBlueGreen`.

Los siguientes temas lo guían en la configuración de una plantilla de CloudFormation para una implementación azul/verde en ECS.

Temas

- [Sobre las implementaciones azul/verde](#)
- [Consideraciones sobre la administración de implementaciones azul/verde de ECS a través de CloudFormation](#)
- [Sintaxis del enlace `AWS::CodeDeploy::BlueGreen`](#)
- [Ejemplo de una plantilla de implementación azul/verde](#)

Sobre las implementaciones azul/verde

En este tema, se ofrece una descripción general del funcionamiento de las implementaciones azul/verde con CloudFormation. También, se explica cómo preparar la plantilla de CloudFormation para las implementaciones azul/verde.

Temas

- [Funcionamiento](#)
- [Actualizaciones de recursos que inician implementaciones verdes](#)
- [Preparación de la plantilla para realizar implementaciones azul/verde de ECS](#)
- [Modelado de la implementación “blue/green” mediante recursos de CloudFormation](#)
- [Conjuntos de cambios](#)
- [Supervisión de los eventos de la pila](#)
- [Permisos de IAM para las implementaciones azul/verde](#)

Funcionamiento

Al usar CloudFormation para realizar implementaciones azul/verde de ECS a través de CodeDeploy, primero se crea una plantilla de pilas que define los recursos de los entornos de aplicaciones azul y verde, incluida la especificación de la configuración de enrutamiento y estabilización del tráfico que se va a utilizar. A continuación, debe crear una pila a partir de esa plantilla. De este modo, generará la aplicación azul (actual). CloudFormation solo crea los recursos azules durante la creación de la pila. Los recursos de las implementaciones «green» no se crean hasta que es necesario.

Posteriormente, si en una actualización futura de una pila se actualiza la definición de tarea o los recursos del conjunto de tareas de la aplicación azul, CloudFormation hará lo siguiente:

- Generará todos los recursos necesarios para el entorno de aplicaciones de verde.
- Desviará el tráfico en función de los parámetros de direccionamiento del tráfico especificados.
- Eliminará los recursos de azul.

Si se produce un error en algún momento antes de que la implementación «green» se realice correctamente y finalice, CloudFormation devolverá la pila al estado que tenía antes de que se iniciara toda la implementación verde.

Actualizaciones de recursos que inician implementaciones verdes

Al realizar una actualización de pila que actualiza una propiedad de recursos de ECS específicos, CloudFormation iniciará un proceso de implementación verde. Los recursos que inician este proceso son los siguientes:

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

Sin embargo, si las actualizaciones de estos recursos no implican cambios en las propiedades que deban reemplazarse, no se iniciará una implementación verde. Para obtener más información, consulte [Comprensión de los comportamientos de actualización de los recursos de la pila](#).

Es importante recordar que no se pueden combinar las actualizaciones de los recursos anteriores con las actualizaciones de otros recursos en la misma operación de actualización de pila. Si necesita actualizar los recursos de la lista y otros recursos de la misma pila, tiene dos opciones:

- Lleve a cabo dos operaciones de actualización de pila diferentes: una que incluya solo las actualizaciones de los recursos anteriores y otra que incluya los cambios de los demás recursos.
- Elimine las secciones `Transform` y `Hooks` de la plantilla y actualice después la pila. En este caso, CloudFormation no realizará una implementación verde.

Preparación de la plantilla para realizar implementaciones azul/verde de ECS

Para habilitar implementaciones azul/verde en la pila, incluya las siguientes secciones en la plantilla de pila antes de realizar la actualización de la pila.

- Agregue una referencia a la transformación `AWS::CodeDeployBlueGreen` en la plantilla:

```
"Transform": [  
  "AWS::CodeDeployBlueGreen"  
],
```

- Agregue una sección `Hooks` que invoque el enlace `AWS::CodeDeploy::BlueGreen` y especifique las propiedades de la implementación. Para obtener más información, consulte [Sintaxis del enlace AWS::CodeDeploy::BlueGreen](#).
- En la sección `Resources`, defina los recursos azul/verde de la implementación.

Puede agregar estas secciones cuando cree por primera vez la plantilla (es decir, antes de crear la pila en sí) o agregarlas a una plantilla existente antes de actualizar la pila. Si especifica la implementación azul-verde para una nueva pila, CloudFormation solo generará los recursos azules durante la creación de la pila; los recursos para la implementación verde no se generarán hasta que lo requiera la actualización de la pila.

Modelado de la implementación “blue/green” mediante recursos de CloudFormation

Para realizar una implementación azul/verde de CodeDeploy en ECS, la plantilla de CloudFormation debe incluir los recursos que van a modelar la implementación, como un equilibrador de carga y un servicio de Amazon ECS. Para obtener más información sobre lo que representan estos recursos, consulte [Antes de empezar una implementación de Amazon ECS](#) en la Guía del usuario de AWS CodeDeploy.

Requisito	Recurso	Obligatorio/opcional	¿Se inicia una implementación azul/verde si se reemplaza?
Clúster de Amazon ECS	AWS::ECS::Cluster	Opcional. Se puede utilizar el clúster predeterminado.	No
Servicio de Amazon ECS	AWS::ECS::Service	Obligatorio.	No
Aplicación o balanceador de carga de red	AWS::ECS::Service LoadBalancer	Obligatorio.	No
Agente de escucha de producción	AWS::ElasticLoadBalancingV2::Listener	Obligatorio.	No
Agente de escucha de prueba	AWS::ElasticLoadBalancingV2::Listener	Opcional.	No
Dos grupos de destino	AWS::ElasticLoadBalancingV2::TargetGroup	Obligatorio.	No

Requisito	Recurso	Obligatorio/opcional	¿Se inicia una implementación azul/verde si se reemplaza?
Definición de tarea de Amazon ECS	AWS::ECS::TaskDefinition	Obligatorio.	Sí
Contenedor de la aplicación de Amazon ECS	AWS::ECS::TaskDefinition Container Definition Name	Obligatorio.	No
Puerto del conjunto de tareas de sustitución	AWS::ECS::TaskDefinition PortMapping ContainerPort	Obligatorio.	No

Conjuntos de cambios

Le recomendamos encarecidamente que cree un conjunto de cambios antes de realizar una actualización de la pila que inicie una implementación verde. Esto le permite ver los cambios reales que se harán en la pila antes de actualizarla. Tenga en cuenta que es posible que los cambios de recursos no aparezcan en el orden en que tendrán lugar. Para obtener más información, consulte [Actualización de pilas de CloudFormation con conjuntos de cambios](#).

Supervisión de los eventos de la pila

Puede ver los eventos de la pila que se generan en cada paso de la implementación de ECS en la pestaña Eventos de la página Pila o mediante la AWS CLI. Para obtener más información, consulte [Monitorizar el progreso de la pila](#).

Permisos de IAM para las implementaciones azul/verde

Para poder CloudFormation realizar correctamente las implementaciones azul/verde, debe tener los siguientes permisos de CodeDeploy:

- `codedeploy:Get*`
- `codedeploy:CreateCloudFormationDeployment`

Para obtener más información, consulte [Acciones, recursos y claves de condición para CodeDeploy](#) en la Referencia de autorizaciones de servicio.

Consideraciones sobre la administración de implementaciones azul/verde de ECS a través de CloudFormation

El proceso de implementación azul/verde de ECS con CloudFormation a través CodeDeploy es diferente al de una implementación estándar de ECS solo con CodeDeploy. Para obtener información detallada sobre estas diferencias, consulte [Diferencias entre implementaciones azul/verde de Amazon ECS a través de CodeDeploy y de CloudFormation](#) en la Guía del usuario de AWS CodeDeploy.

Al administrar su implementación azul/verde con CloudFormation, hay ciertas limitaciones y consideraciones que debe tener en cuenta:

- Solo las actualizaciones de determinados recursos iniciarán una implementación verde. Para obtener más información, consulte [Actualizaciones de recursos que inician implementaciones verdes](#).
- No puede incluir actualizaciones de recursos que inicien implementaciones verde y actualizaciones de otros recursos en la misma actualización de pila. Para obtener más información, consulte [Actualizaciones de recursos que inician implementaciones verdes](#).
- Solo puede especificar un único servicio de ECS como destino de implementación.
- Los parámetros cuyos valores están enmascarados por CloudFormation no pueden actualizarse a través de CodeDeploy durante una implementación verde, ya que se produciría un error y la actualización de la pila no se realizaría correctamente. Entre ellos se incluyen:
 - Parámetros definidos con el atributo NoEcho.
 - Parámetros que utilizan referencias dinámicas para recuperar sus valores a partir de servicios externos. Para obtener más información sobre las referencias dinámicas, consulte [Obtención de valores almacenados en otros servicios con referencias dinámicas](#).
- Para cancelar una implementación verde que aún está en curso, cancele la actualización de pila en CloudFormation, no CodeDeploy o ECS. Para obtener más información, consulte [Cancelar una actualización de pila](#). Una vez finalizada la actualización, no podrá cancelarla. No obstante, puede volver a actualizar una pila con una configuración anterior.
- Las siguientes características de CloudFormation no son compatibles actualmente con plantillas que definen las implementaciones azul/verde de ECS:
 - Declaración de [salidas](#) o uso de [Fn::ImportValue](#) para importar valores de otras pilas.

- Importación de recursos. Para obtener más información sobre la importación de recursos, consulte [Cómo importar recursos de AWS a una pila de CloudFormation](#).
- Uso del enlace `AWS::CodeDeploy::BlueGreen` en una plantilla que incluya recursos de pila anidados. Para obtener más información sobre las pilas anidadas, consulte [División de una plantilla en piezas reutilizables utilizando pilas anidadas](#).
- Uso del enlace `AWS::CodeDeploy::BlueGreen` en una pila anidada.

Sintaxis del enlace **AWS::CodeDeploy::BlueGreen**

La siguiente sintaxis describe la estructura de un enlace `AWS::CodeDeploy::BlueGreen` para las implementaciones azul/verde de ECS.

Sintaxis

```
"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)"
      ]
    }
  }
}

```

Propiedades

ID lógico (también denominado nombre lógico)

El ID lógico de un enlace declarado en la sección Hooks de la plantilla. El ID lógico tiene que ser alfanumérico (A-Za-z0-9) y único dentro de la plantilla.

Obligatorio: sí

Type

El tipo de enlace. `AWS::CodeDeploy::BlueGreen`

Obligatorio: sí

Properties

Propiedades del enlace.

Obligatorio: sí

TrafficRoutingConfig

Configuración del direccionamiento del tráfico.

Obligatorio: no

La configuración predeterminada desvía el tráfico de valor controlado en función del tiempo, con un porcentaje de pasos del 15 % y un tiempo de procesamiento de cinco minutos.

Type

Tipo de desvío de tráfico que se utiliza en la configuración de la implementación.

Valores válidos: `AllAtOnce` | `TimeBasedCanary` | `TimeBasedLinear`

Obligatorio: sí

TimeBasedCanary

Especifica una configuración que desvía el tráfico de una versión de la implementación a otra en dos incrementos.

Obligatorio: condicional. Si especifica `TimeBasedCanary` como el tipo de enrutamiento de tráfico, debe incluir el parámetro `TimeBasedCanary`.

StepPercentage

Porcentaje del tráfico que se va a desviar en el primer incremento de una implementación `TimeBasedCanary`. El porcentaje de pasos debe ser igual o superior al 14 %.

Obligatorio: no

BakeTimeMins

Número de minutos entre el primer y el segundo desvío de tráfico de una implementación `TimeBasedCanary`.

Obligatorio: no

TimeBasedLinear

Especifica una configuración que desvía el tráfico de una versión de la implementación a otra en incrementos iguales, con el mismo número de minutos entre cada incremento.

Obligatorio: condicional. Si especifica `TimeBasedLinear` como el tipo de enrutamiento de tráfico, debe incluir el parámetro `TimeBasedLinear`.

StepPercentage

Porcentaje de tráfico que se desvía al comienzo de cada incremento de una implementación `TimeBasedLinear`. El porcentaje de pasos debe ser igual o superior al 14 %.

Obligatorio: no

BakeTimeMins

Número de minutos entre cada desvío del tráfico incremental en una implementación `TimeBasedLinear`.

Obligatorio: no

AdditionalOptions

Otras opciones adicionales de la implementación «blue/green».

Obligatorio: no

TerminationWaitTimeInMinutes

Especifica el tiempo de espera, en minutos, antes de finalizar los recursos de azul.

Obligatorio: no

LifecycleEventHooks

Utilice los enlaces de eventos del ciclo de vida para especificar una función de Lambda a la que CodeDeploy puede llamar para validar una implementación. Puede utilizar la misma

función u otra distinta para los eventos del ciclo de vida de la implementación. Cuando finalicen las pruebas de validación, la función de Lambda `AfterAllowTraffic` vuelve a llamar a CodeDeploy y devuelve como resultado `Succeeded` o `Failed`. Para obtener más información, consulte la [Sección “Enlaces” de AppSpec](#) en la Guía del usuario de AWS CodeDeploy.

Obligatorio: no

`BeforeInstall`

Función que se utiliza para ejecutar tareas antes de crear el conjunto de tareas de sustitución.

Obligatorio: no

`AfterInstall`

Función que se utiliza para ejecutar tareas una vez que el conjunto de tareas se ha creado y uno de los grupos de destino se ha asociado con él.

Obligatorio: no

`AfterAllowTestTraffic`

Función que se utiliza para ejecutar tareas después de que el agente de escucha de prueba envía tráfico al conjunto de tareas de sustitución.

Obligatorio: no

`BeforeAllowTraffic`

Función que se utiliza para ejecutar tareas una vez que el segundo grupo de destino se ha asociado con el conjunto de tareas de sustitución, pero antes de que se envíe tráfico al conjunto de tareas de sustitución.

Obligatorio: no

`AfterAllowTraffic`

Función que se utiliza para ejecutar tareas después de que el segundo grupo de destino envíe tráfico al conjunto de tareas de sustitución.

Obligatorio: no

ServiceRole

Rol de ejecución que se utiliza en CloudFormation para realizar implementaciones azul/verde. Para obtener una lista de los permisos necesarios, consulte [Permisos de IAM para las implementaciones azul/verde](#).

Obligatorio: no

Applications

Especifica las propiedades de la aplicación Amazon ECS.

Obligatorio: sí

Target

Obligatorio: sí

Type

El tipo de recurso.

Obligatorio: sí

LogicalID

ID lógico del recurso.

Obligatorio: sí

ECSAttributes

Recursos que representan los diferentes requisitos de la implementación de la aplicación Amazon ECS.

Obligatorio: sí

TaskDefinitions

ID lógico del recurso [AWS::ECS::TaskDefinition](#) que va a ejecutar el contenedor Docker que contiene la aplicación Amazon ECS.

Obligatorio: sí

TaskSets

Identificadores lógicos de los recursos [AWS::ECS::TaskSet](#) que se van a utilizar como conjuntos de tareas en la aplicación.

Obligatorio: sí

TrafficRouting

Especifica los recursos que se utilizan para el direccionamiento del tráfico.

Obligatorio: sí

ProdTrafficRoute

Agente de escucha que se va a utilizar en el balanceador de carga para dirigir el tráfico hacia los grupos de destino.

Obligatorio: sí

Type

El tipo del recurso. `AWS::ElasticLoadBalancingV2::Listener`

Obligatorio: sí

LogicalID

ID lógico del recurso.

Obligatorio: sí

TestTrafficRoute

Agente de escucha que se va a utilizar en el balanceador de carga para dirigir el tráfico hacia los grupos de destino.

Obligatorio: sí

Type

El tipo del recurso. `AWS::ElasticLoadBalancingV2::Listener`

Obligatorio: sí

LogicalID

ID lógico del recurso.

Obligatorio: no

TargetGroups

ID lógico de los recursos que se van a utilizar como grupos de destino para dirigir el tráfico al destino registrado.

Obligatorio: sí

Ejemplo de una plantilla de implementación azul/verde

En la siguiente plantilla de ejemplo, se configura una implementación azul/verde de CodeDeploy en ECS, con un progreso de direccionamiento del tráfico del 15 % en cada paso y un periodo de estabilización de 5 minutos entre cada paso.

Si se crea una pila con la plantilla, se aprovisionará la configuración inicial de la implementación. Si después se hace algún cambio en las propiedades del recurso BlueTaskSet que requiera que este se reemplace, CloudFormation iniciará una implementación verde como parte de la actualización de la pila.

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Parameters": {
    "Vpc": { "Type": "AWS::EC2::VPC::Id" },
    "Subnet1": { "Type": "AWS::EC2::Subnet::Id" },
    "Subnet2": { "Type": "AWS::EC2::Subnet::Id" }
  },
  "Transform": [ "AWS::CodeDeployBlueGreen" ],
  "Hooks": {
    "CodeDeployBlueGreenHook": {
      "Type": "AWS::CodeDeploy::BlueGreen",
      "Properties": {
        "TrafficRoutingConfig": {
          "Type": "TimeBasedCanary",
          "TimeBasedCanary": {
            "StepPercentage": 15,
            "BakeTimeMins": 5
          }
        }
      }
    },
    "Applications": [
      {
        "Target": {
          "Type": "AWS::ECS::Service",
          "LogicalID": "ECSDemoService"
        },
        "ECSAttributes": {
          "TaskDefinitions": [ "BlueTaskDefinition", "GreenTaskDefinition" ],
```

```

        "TaskSets": [ "BlueTaskSet", "GreenTaskSet" ],
        "TrafficRouting": {
            "ProdTrafficRoute": {
                "Type": "AWS::ElasticLoadBalancingV2::Listener",
                "LogicalID": "ALBListenerProdTraffic"
            },
            "TargetGroups": [ "ALBTargetGroupBlue", "ALBTargetGroupGreen" ]
        }
    }
}
],
}
},
"Resources": {
    "ExampleSecurityGroup": {
        "Type": "AWS::EC2::SecurityGroup",
        "Properties": {
            "GroupDescription": "Security group for ec2 access",
            "VpcId": { "Ref": "Vpc" },
            "SecurityGroupIngress": [
                {
                    "IpProtocol": "tcp",
                    "FromPort": 80,
                    "ToPort": 80,
                    "CidrIp": "0.0.0.0/0"
                },
                {
                    "IpProtocol": "tcp",
                    "FromPort": 8080,
                    "ToPort": 8080,
                    "CidrIp": "0.0.0.0/0"
                },
                {
                    "IpProtocol": "tcp",
                    "FromPort": 22,
                    "ToPort": 22,
                    "CidrIp": "0.0.0.0/0"
                }
            ]
        }
    },
    "ALBTargetGroupBlue": {
        "Type": "AWS::ElasticLoadBalancingV2::TargetGroup",

```

```

    "Properties":{
      "HealthCheckIntervalSeconds":5,
      "HealthCheckPath":"/",
      "HealthCheckPort":"80",
      "HealthCheckProtocol":"HTTP",
      "HealthCheckTimeoutSeconds":2,
      "HealthyThresholdCount":2,
      "Matcher":{ "HttpCode":"200" },
      "Port":80,
      "Protocol":"HTTP",
      "Tags":[{ "Key":"Group","Value":"Example" }],
      "TargetType":"ip",
      "UnhealthyThresholdCount":4,
      "VpcId":{ "Ref":"Vpc" }
    }
  },
  "ALBTargetGroupGreen":{
    "Type":"AWS::ElasticLoadBalancingV2::TargetGroup",
    "Properties":{
      "HealthCheckIntervalSeconds":5,
      "HealthCheckPath":"/",
      "HealthCheckPort":"80",
      "HealthCheckProtocol":"HTTP",
      "HealthCheckTimeoutSeconds":2,
      "HealthyThresholdCount":2,
      "Matcher":{ "HttpCode":"200" },
      "Port":80,
      "Protocol":"HTTP",
      "Tags":[{ "Key":"Group","Value":"Example" }],
      "TargetType":"ip",
      "UnhealthyThresholdCount":4,
      "VpcId":{ "Ref":"Vpc" }
    }
  },
  "ExampleALB":{
    "Type":"AWS::ElasticLoadBalancingV2::LoadBalancer",
    "Properties":{
      "Scheme":"internet-facing",
      "SecurityGroups":[{ "Ref":"ExampleSecurityGroup" }],
      "Subnets":[{ "Ref":"Subnet1" },{ "Ref":"Subnet2" }],
      "Tags":[{ "Key":"Group","Value":"Example" }],
      "Type":"application",
      "IpAddressType":"ipv4"
    }
  }
}

```

```

},
"ALBListenerProdTraffic":{
  "Type":"AWS::ElasticLoadBalancingV2::Listener",
  "Properties":{
    "DefaultActions":[
      {
        "Type":"forward",
        "ForwardConfig":{
          "TargetGroups":[
            {
              "TargetGroupArn":{ "Ref":"ALBTargetGroupBlue" },
              "Weight":1
            }
          ]
        }
      }
    ],
    "LoadBalancerArn":{ "Ref":"ExampleALB" },
    "Port":80,
    "Protocol":"HTTP"
  }
},
"ALBListenerProdRule":{
  "Type":"AWS::ElasticLoadBalancingV2::ListenerRule",
  "Properties":{
    "Actions":[
      {
        "Type":"forward",
        "ForwardConfig":{
          "TargetGroups":[
            {
              "TargetGroupArn":{ "Ref":"ALBTargetGroupBlue" },
              "Weight":1
            }
          ]
        }
      }
    ],
    "Conditions":[
      {
        "Field":"http-header",
        "HttpHeaderConfig":{
          "HttpHeaderName":"User-Agent",
          "Values":[ "Mozilla" ]
        }
      }
    ]
  }
}

```



```

        }
      }
    ],
    "ListenerArn":{ "Ref":"ALBListenerProdTraffic" },
    "Priority":1
  }
},
"ECSTaskExecutionRole":{
  "Type":"AWS::IAM::Role",
  "Properties":{
    "AssumeRolePolicyDocument":{
      "Version": "2012-10-17",
      "Statement":[
        {
          "Sid":"",
          "Effect":"Allow",
          "Principal":{
            "Service":"ecs-tasks.amazonaws.com"
          },
          "Action":"sts:AssumeRole"
        }
      ]
    },
    "ManagedPolicyArns":[ "arn:aws:iam::aws:policy/service-role/AmazonECSTaskExecutionRolePolicy" ]
  }
},
"BlueTaskDefinition":{
  "Type":"AWS::ECS::TaskDefinition",
  "Properties":{
    "ExecutionRoleArn":{
      "Fn::GetAtt":[ "ECSTaskExecutionRole","Arn" ]
    },
    "ContainerDefinitions":[
      {
        "Name":"DemoApp",
        "Image":"nginxdemos/hello:latest",
        "Essential":true,
        "PortMappings":[
          {
            "HostPort":80,
            "Protocol":"tcp",
            "ContainerPort":80
          }
        ]
      }
    ]
  }
}

```

```

        ]
      }
    ],
    "RequiresCompatibilities":[ "FARGATE" ],
    "NetworkMode":"awsvpc",
    "Cpu":"256",
    "Memory":"512",
    "Family":"ecs-demo"
  }
},
"ECSDemoCluster":{
  "Type":"AWS::ECS::Cluster",
  "Properties":{}
},
"ECSDemoService":{
  "Type":"AWS::ECS::Service",
  "Properties":{
    "Cluster":{ "Ref":"ECSDemoCluster" },
    "DesiredCount":1,
    "DeploymentController":{ "Type":"EXTERNAL" }
  }
},
"BlueTaskSet":{
  "Type":"AWS::ECS::TaskSet",
  "Properties":{
    "Cluster":{ "Ref":"ECSDemoCluster" },
    "LaunchType":"FARGATE",
    "NetworkConfiguration":{
      "AwsVpcConfiguration":{
        "AssignPublicIp":"ENABLED",
        "SecurityGroups":[{ "Ref":"ExampleSecurityGroup" }],
        "Subnets":[{ "Ref":"Subnet1" },{ "Ref":"Subnet2" }]
      }
    }
  },
  "PlatformVersion":"1.4.0",
  "Scale":{
    "Unit":"PERCENT",
    "Value":100
  },
  "Service":{ "Ref":"ECSDemoService"},
  "TaskDefinition":{ "Ref":"BlueTaskDefinition" },
  "LoadBalancers":[
    {
      "ContainerName":"DemoApp",

```

```

        "ContainerPort":80,
        "TargetGroupArn":{ "Ref":"ALBTargetGroupBlue" }
    }
]
}
},
"PrimaryTaskSet":{
    "Type":"AWS::ECS::PrimaryTaskSet",
    "Properties":{
        "Cluster":{ "Ref":"ECSDemoCluster" },
        "Service":{ "Ref":"ECSDemoService" },
        "TaskSetId":{ "Fn::GetAtt":[ "BlueTaskSet","Id" ]
    }
}
}
}
}
}

```

YAML

```

AWSTemplateFormatVersion: 2010-09-09
Parameters:
  Vpc:
    Type: AWS::EC2::VPC::Id
  Subnet1:
    Type: AWS::EC2::Subnet::Id
  Subnet2:
    Type: AWS::EC2::Subnet::Id
Transform:
  - 'AWS::CodeDeployBlueGreen'
Hooks:
  CodeDeployBlueGreenHook:
    Type: AWS::CodeDeploy::BlueGreen
    Properties:
      TrafficRoutingConfig:
        Type: TimeBasedCanary
        TimeBasedCanary:
          StepPercentage: 15
          BakeTimeMins: 5
      Applications:
        - Target:
            Type: AWS::ECS::Service
            LogicalID: ECSDemoService

```

```
ECSAttributes:
  TaskDefinitions:
    - BlueTaskDefinition
    - GreenTaskDefinition
  TaskSets:
    - BlueTaskSet
    - GreenTaskSet
  TrafficRouting:
    ProdTrafficRoute:
      Type: AWS::ElasticLoadBalancingV2::Listener
      LogicalID: ALBListenerProdTraffic
    TargetGroups:
      - ALBTargetGroupBlue
      - ALBTargetGroupGreen
```

Resources:

```
ExampleSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Security group for ec2 access
    VpcId: !Ref Vpc
    SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: 80
        ToPort: 80
        CidrIp: 0.0.0.0/0
      - IpProtocol: tcp
        FromPort: 8080
        ToPort: 8080
        CidrIp: 0.0.0.0/0
      - IpProtocol: tcp
        FromPort: 22
        ToPort: 22
        CidrIp: 0.0.0.0/0
```

```
ALBTargetGroupBlue:
  Type: AWS::ElasticLoadBalancingV2::TargetGroup
  Properties:
    HealthCheckIntervalSeconds: 5
    HealthCheckPath: /
    HealthCheckPort: '80'
    HealthCheckProtocol: HTTP
    HealthCheckTimeoutSeconds: 2
    HealthyThresholdCount: 2
    Matcher:
      HttpCode: '200'
```

```
    Port: 80
    Protocol: HTTP
    Tags:
      - Key: Group
        Value: Example
    TargetType: ip
    UnhealthyThresholdCount: 4
    VpcId: !Ref Vpc
ALBTargetGroupGreen:
  Type: AWS::ElasticLoadBalancingV2::TargetGroup
  Properties:
    HealthCheckIntervalSeconds: 5
    HealthCheckPath: /
    HealthCheckPort: '80'
    HealthCheckProtocol: HTTP
    HealthCheckTimeoutSeconds: 2
    HealthyThresholdCount: 2
    Matcher:
      HttpCode: '200'
    Port: 80
    Protocol: HTTP
    Tags:
      - Key: Group
        Value: Example
    TargetType: ip
    UnhealthyThresholdCount: 4
    VpcId: !Ref Vpc
ExampleALB:
  Type: AWS::ElasticLoadBalancingV2::LoadBalancer
  Properties:
    Scheme: internet-facing
    SecurityGroups:
      - !Ref ExampleSecurityGroup
    Subnets:
      - !Ref Subnet1
      - !Ref Subnet2
    Tags:
      - Key: Group
        Value: Example
    Type: application
    IpAddressType: ipv4
ALBListenerProdTraffic:
  Type: AWS::ElasticLoadBalancingV2::Listener
  Properties:
```

```
DefaultActions:
  - Type: forward
    ForwardConfig:
      TargetGroups:
        - TargetGroupArn: !Ref ALBTargetGroupBlue
          Weight: 1
    LoadBalancerArn: !Ref ExampleALB
    Port: 80
    Protocol: HTTP
ALBListenerProdRule:
  Type: AWS::ElasticLoadBalancingV2::ListenerRule
  Properties:
    Actions:
      - Type: forward
        ForwardConfig:
          TargetGroups:
            - TargetGroupArn: !Ref ALBTargetGroupBlue
              Weight: 1
    Conditions:
      - Field: http-header
        HttpHeaderConfig:
          HttpHeaderName: User-Agent
          Values:
            - Mozilla
    ListenerArn: !Ref ALBListenerProdTraffic
    Priority: 1
ECSTaskExecutionRole:
  Type: AWS::IAM::Role
  Properties:
    AssumeRolePolicyDocument:
      Version: 2012-10-17
      Statement:
        - Sid: ''
          Effect: Allow
          Principal:
            Service: ecs-tasks.amazonaws.com
          Action: 'sts:AssumeRole'
    ManagedPolicyArns:
      - 'arn:aws:iam::aws:policy/service-role/AmazonECSTaskExecutionRolePolicy'
BlueTaskDefinition:
  Type: AWS::ECS::TaskDefinition
  Properties:
    ExecutionRoleArn: !GetAtt
      - ECSTaskExecutionRole
```

```
- Arn
ContainerDefinitions:
  - Name: DemoApp
    Image: 'nginxdemos/hello:latest'
    Essential: true
    PortMappings:
      - HostPort: 80
        Protocol: tcp
        ContainerPort: 80
    RequiresCompatibilities:
      - FARGATE
    NetworkMode: awsvpc
    Cpu: '256'
    Memory: '512'
    Family: ecs-demo
ECSDemoCluster:
  Type: AWS::ECS::Cluster
  Properties: {}
ECSDemoService:
  Type: AWS::ECS::Service
  Properties:
    Cluster: !Ref ECSDemoCluster
    DesiredCount: 1
    DeploymentController:
      Type: EXTERNAL
BlueTaskSet:
  Type: AWS::ECS::TaskSet
  Properties:
    Cluster: !Ref ECSDemoCluster
    LaunchType: FARGATE
    NetworkConfiguration:
      AwsVpcConfiguration:
        AssignPublicIp: ENABLED
        SecurityGroups:
          - !Ref ExampleSecurityGroup
        Subnets:
          - !Ref Subnet1
          - !Ref Subnet2
    PlatformVersion: 1.4.0
    Scale:
      Unit: PERCENT
      Value: 100
    Service: !Ref ECSDemoService
    TaskDefinition: !Ref BlueTaskDefinition
```

```
LoadBalancers:
  - ContainerName: DemoApp
    ContainerPort: 80
    TargetGroupArn: !Ref ALBTargetGroupBlue
PrimaryTaskSet:
  Type: AWS::ECS::PrimaryTaskSet
  Properties:
    Cluster: !Ref ECSDemoCluster
    Service: !Ref ECSDemoService
    TaskSetId: !GetAtt
      - BlueTaskSet
      - Id
```

Fragmentos de plantillas de CloudFormation

Esta sección proporciona una serie de situaciones de ejemplo que puede utilizar para comprender cómo declarar diferentes partes de la plantilla de CloudFormation. También puede utilizar los fragmentos de código como punto de partida para las secciones de sus plantillas personalizadas.

Note

Dado que las plantillas de CloudFormation deben ser compatibles con JSON, no se prevé un carácter de continuación de línea. El ajuste de los fragmentos en este documento puede ser aleatorio si la línea tiene más de 68 caracteres.

Temas

- [Fragmentos de código de plantillas generales](#)
- [Fragmentos de código de plantillas de CloudFormation de escalado automático](#)
- [Fragmentos de plantillas de la consola de facturación de AWS](#)
- [CloudFormationFragmentos de plantillas de](#)
- [Fragmentos de código de plantillas de Amazon CloudFront](#)
- [Fragmentos de código de plantillas de Amazon CloudWatch](#)
- [Fragmentos de la plantilla de Amazon CloudWatch Logs](#)
- [Fragmentos de plantillas de Amazon DynamoDB](#)
- [Fragmentos de plantillas de Amazon EC2 CloudFormation](#)

- [Ejemplos de plantillas de Amazon Elastic Container Service](#)
- [Plantilla de ejemplo de Amazon Elastic File System](#)
- [Fragmentos de código de plantillas de Elastic Beanstalk](#)
- [Fragmentos de plantillas de Elastic Load Balancing](#)
- [AWS Identity and Access Management](#) Fragmentos de plantillas de
- [AWS Lambda](#) Plantilla de
- [Fragmentos de plantillas de Amazon Redshift](#)
- [Fragmentos de las plantillas de Amazon RDS](#)
- [Fragmentos de plantilla de Route 53](#)
- [Fragmentos de código de plantillas de Amazon S3](#)
- [Fragmentos de plantillas de Amazon SNS](#)
- [Fragmentos de código de plantillas de Amazon SQS](#)
- [Fragmentos de plantilla de Amazon Timestream](#)

Fragmentos de código de plantillas generales

Los siguientes ejemplos muestran diferentes características de plantilla de CloudFormation que no son específicas de un servicio de AWS.

Temas

- [Propiedad UserData codificada en Base64](#)
- [Propiedad UserData codificada en Base64 con AccessKey y SecretKey](#)
- [Sección Parameters con un parámetro de cadena literal](#)
- [Sección Parameters con parámetro de cadena con limitación de expresiones regulares](#)
- [Sección Parameters con parámetro numérico con restricciones MinValue y MaxValue](#)
- [Sección Parameters con parámetro numérico con restricción AllowedValues](#)
- [Sección Parameters con un parámetro CommaDelimitedList literal](#)
- [Sección Parameters con valor de parámetros basado en pseudoparámetro](#)
- [Sección Mapping con tres asignaciones](#)
- [Description basada en cadena literal](#)

- [Sección Outputs con una salida de cadena literal](#)
- [Sección Outputs con una referencia de recursos y una salida de pseudoreferencia](#)
- [Sección Outputs con una salida basada en una función, una cadena literal, una referencia y un pseudoparámetro](#)
- [Versión de formato de plantilla](#)
- [Propiedad Tags de AWS](#)

Propiedad UserData codificada en Base64

Este ejemplo muestra el conjunto de una propiedad UserData que utiliza las funciones Fn::Base64 y Fn::Join. Las referencias MyValue y MyName son parámetros que deben definirse en la sección Parameters de la plantilla. La cadena literal Hello World es simplemente otro valor que este ejemplo introduce como parte de los UserData.

JSON

```
"UserData" : {
  "Fn::Base64" : {
    "Fn::Join" : [ ",", [
      { "Ref" : "MyValue" },
      { "Ref" : "MyName" },
      "Hello World" ] ]
  }
}
```

YAML

```
UserData:
  Fn::Base64: !Sub |
    Ref: MyValue
    Ref: MyName
    Hello World
```

Propiedad UserData codificada en Base64 con AccessKey y SecretKey

Este ejemplo muestra el conjunto de una propiedad UserData que utiliza las funciones Fn::Base64 y Fn::Join. Incluye la información AccessKey y SecretKey. Las referencias

AccessKey y SecretKey son parámetros que deben definirse en la sección Parameters de la plantilla.

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}
```

Sección Parameters con un parámetro de cadena literal

El siguiente ejemplo muestra una declaración de la sección Parameters (Parámetros) válida en la que se declara un parámetro de tipo String individual.

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

Sección Parameters con parámetro de cadena con limitación de expresiones regulares

El siguiente ejemplo muestra una declaración de la sección Parameters (Parámetros) válida en la que se declara un parámetro de tipo String individual. El parámetro AdminUserAccount tiene el valor predeterminado de admin. El valor del parámetro debe tener una longitud mínima de 1, una longitud máxima de 16 y contener caracteres alfabéticos y números, pero debe comenzar por un carácter alfabético.

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]*'
```

Sección Parameters con parámetro numérico con restricciones MinValue y MaxValue

El siguiente ejemplo muestra una declaración de la sección Parameters (Parámetros) válida en la que se declara un parámetro de tipo Number individual. El parámetro WebServerPort tiene un valor predeterminado de 80, un valor mínimo de 1 y un valor máximo de 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
```

Sección Parameters con parámetro numérico con restricción AllowedValues

El siguiente ejemplo muestra una declaración de la sección Parameters (Parámetros) válida en la que se declara un parámetro de tipo Number individual. El parámetro WebServerPort tiene un valor predeterminado de 80 y solo permite valores de 80 y 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
```

Sección Parameters con un parámetro CommaDelimitedList literal

El siguiente ejemplo muestra una declaración de la sección Parameters válida en la que se declara un parámetro de tipo CommaDelimitedList individual. La propiedad NoEcho se establece en TRUE, lo que enmascara su valor con asteriscos (*****) en la salida describe-stacks, excepto para la información almacenada en las ubicaciones especificadas a continuación.

Important

El uso del atributo NoEcho no enmascara ninguna información almacenada en lo que se muestra a continuación:

- La sección de la plantilla Metadata. CloudFormation no transforma, modifica ni redacta ninguna información que incluya en la sección Metadata. Para obtener más información, consulte [Metadata](#).
- La sección de la plantilla Outputs. Para obtener más información, consulte [Salidas](#).
- El atributo Metadata de una definición de recurso. Para obtener más información, consulte [Metadata atributo](#).

Recomendamos encarecidamente que no utilice estos mecanismos para incluir información confidencial, como contraseñas o secretos.

Important

En lugar de integrar información confidencial directamente en las plantillas de CloudFormation, se recomienda utilizar parámetros dinámicos en la plantilla de la pila para hacer referencia a la información confidencial almacenada y administrada fuera de CloudFormation, como en AWS Systems Manager Parameter Store o AWS Secrets Manager.

Para obtener más información, consulte la práctica recomendada [No integre credenciales en sus plantillas](#).

JSON

```
"Parameters" : {
  "UserRoles" : {
    "Type" : "CommaDelimitedList",
    "Default" : "guest,newhire",
    "NoEcho" : "TRUE"
  }
}
```

YAML

```
Parameters:
  UserRoles:
    Type: CommaDelimitedList
    Default: "guest,newhire"
    NoEcho: true
```

Sección Parameters con valor de parámetros basado en pseudoparámetro

El siguiente ejemplo muestra comandos en los datos de usuario de EC2 que utilizan los pseudoparámetros `AWS::StackName` y `AWS::Region`. Para obtener más información sobre pseudoparámetros, consulte . [Obtención de valores de AWS mediante pseudoparámetros](#).

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}

```

Sección Mapping con tres asignaciones

El siguiente ejemplo muestra una declaración de sección Mapping válida que contiene tres asignaciones. La asignación, cuando coincide con una clave de mapeo de Stop, SlowDown o Go, proporciona los valores RGB asignados al atributo RGBColor correspondiente.

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 basada en cadena literal

El siguiente ejemplo muestra una declaración de sección `Description` válida donde el valor se basa en una cadena literal. Este fragmento de código puede ser para plantillas, parámetros, recursos, propiedades o salidas.

JSON

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

YAML

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

Sección Outputs con una salida de cadena literal

Este ejemplo muestra una asignación de salida basada en una cadena 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
```

Sección Outputs con una referencia de recursos y una salida de pseudoreferencia

Este ejemplo muestra una sección Outputs con dos asignaciones de salida. Una se basa en un recurso y la otra se basa en una pseudoreferencia.

JSON

```
"Outputs" : {
  "SNSTopic" : { "Value" : { "Ref" : "MyNotificationTopic" } },
  "StackName" : { "Value" : { "Ref" : "AWS::StackName" } }
}
```

YAML

```
Outputs:
  SNSTopic:
    Value: !Ref MyNotificationTopic
  StackName:
    Value: !Ref AWS::StackName
```

Sección Outputs con una salida basada en una función, una cadena literal, una referencia y un pseudoparámetro

Este ejemplo muestra una sección Outputs con una asignación de salida. La función Join se utiliza para concatenar el valor con un signo de porcentaje como delimitador.

JSON

```
"Outputs" : {
  "MyOutput" : {
    "Value" : { "Fn::Join" :
      [ "%", [ "A-string", {"Ref" : "AWS::StackName" } ] ]
    }
  }
}
```

```
    }  
  }  
}
```

YAML

```
Outputs:  
  MyOutput:  
    Value: !Join [ %, [ 'A-string', !Ref 'AWS::StackName' ]]
```

Versión de formato de plantilla

El siguiente fragmento de código muestra una declaración válida de sección `AWSTemplateFormatVersion`.

JSON

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

YAML

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

Propiedad Tags de AWS

En este ejemplo se muestra una propiedad `Tags` de AWS. Esta propiedad se especifica dentro de la sección `Properties` de un recurso. Cuando se crea el recurso, se etiqueta con las etiquetas que usted declare.

JSON

```
"Tags" : [  
  {  
    "Key" : "keyname1",  
    "Value" : "value1"  
  },  
  {  
    "Key" : "keyname2",  
    "Value" : "value2"  
  }  
]
```

```
}  
]
```

YAML

```
Tags:  
-  
  Key: "keyname1"  
  Value: "value1"  
-  
  Key: "keyname2"  
  Value: "value2"
```

Fragmentos de código de plantillas de CloudFormation de escalado automático

Amazon EC2 Auto Scaling le permite escalar de manera automática instancias de Amazon EC2, ya sea con políticas de escalado o con políticas programadas de escalado. Los grupos de escalado automático son colecciones de instancias de Amazon EC2 que habilitan el escalado automático y las características de administración de flotas, como las políticas de escalado, las acciones programadas, las comprobaciones de estado, los enlaces de ciclo de vida y el equilibrio de cargas.

Application Auto Scaling proporciona escalado automático de distintos recursos más allá de Amazon EC2, ya sea con políticas de escalado o con escalado programado.

Puede crear y configurar grupos de escalado automático, políticas de escalado, acciones programadas y otros recursos de escalado automático como parte de su infraestructura mediante plantillas CloudFormation. Las plantillas facilitan la administración y la automatización de la implementación de los recursos de escalado automático de forma repetible y coherente.

Los siguientes fragmentos de plantilla de ejemplo describen los recursos o componentes de CloudFormation para Amazon EC2 Auto Scaling y el escalado automático de aplicaciones. Estos fragmentos están diseñados para integrarse en una plantilla y no están pensados para ejecutarse de forma independiente.

Categorías de fragmentos

- [Configuración de los recursos de Amazon EC2 Auto Scaling con CloudFormation](#)
- [Configuración de recursos de escalado automático de aplicaciones con CloudFormation](#)

Configuración de los recursos de Amazon EC2 Auto Scaling con CloudFormation

En los ejemplos siguientes se muestran diferentes fragmentos para incluir en las plantillas para su uso con Amazon EC2 Auto Scaling.

Categorías de fragmentos

- [Creación de un grupo de escalado automático de instancia única](#)
- [Creación de un grupo de escalado automático con un equilibrador de carga adjunto](#)
- [Creación de un grupo de escalado automático con notificaciones](#)
- [Creación de un grupo de escalado automático que utiliza una CreationPolicy y una UpdatePolicy](#)
- [Creación de una política de escalado por pasos](#)
- [Ejemplos del grupo de instancias mixtas](#)
- [Ejemplos de configuración de lanzamiento](#)

Creación de un grupo de escalado automático de instancia única

En este ejemplo se muestra un recurso de [AWS::AutoScaling::AutoScalingGroup](#) con una sola instancia para ayudarlo a comenzar. La propiedad `VPCZoneIdentifier` del grupo de escalado automático especifica una lista de subredes existentes en tres zonas de disponibilidad diferentes. Debe especificar los ID de subredes aplicables de su cuenta antes de crear la pila. La propiedad `LaunchTemplate` hace referencia a un recurso [AWS::EC2::LaunchTemplate](#) con el nombre lógico `myLaunchTemplate` definido en cualquier otro lugar de la plantilla.

Note

Para obtener más ejemplos de plantillas de lanzamiento, consulte [Creación de plantillas de inicialización con CloudFormation](#) en la sección de fragmentos de Amazon EC2 y en la sección [Ejemplos](#) en el recurso de `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'

```

Creación de un grupo de escalado automático con un equilibrador de carga adjunto

En este ejemplo se muestra un recurso de [AWS::AutoScaling::AutoScalingGroup](#) para el equilibrador de carga en varios servidores. Especifica los nombres lógicos de los recursos de AWS declarados en otra parte de la misma plantilla.

1. La propiedad `VPCZoneIdentifier` especifica los nombres lógicos de dos recursos de [AWS::EC2::Subnet](#) donde se crearán las instancias del grupo de escalado automático EC2: `myPublicSubnet1` y `myPublicSubnet2`.
2. La propiedad `LaunchTemplate` especifica un recurso de [AWS::EC2::LaunchTemplate](#) con el nombre lógico `myLaunchTemplate`.

3. La propiedad `TargetGroupARNs` muestra los grupos de destino para Balanceador de carga de aplicaciones o Balanceador de carga de red utilizados para enrutar tráfico al grupo de Auto Scaling. En este ejemplo, se especifica un grupo de destino, que es un recurso de [AWS::ElasticLoadBalancingV2::TargetGroup](#) con el nombre lógico `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
```

Véase también

Para obtener un ejemplo detallado que crea un grupo de escalado automático con una política de escalado de seguimiento de destino basada en la métrica predefinida de `ALBRequestCountPerTarget` para el equilibrador de carga de aplicación, consulte la sección [Ejemplos](#) en el recurso de `AWS::AutoScaling::ScalingPolicy`.

Creación de un grupo de escalado automático con notificaciones

En este ejemplo se muestra un recurso de [AWS::AutoScaling::AutoScalingGroup](#) que envía notificaciones de Amazon SNS cuando se producen los eventos especificados. La propiedad `NotificationConfigurations` especifica el tema de SNS donde CloudFormation envía una notificación y los eventos que causarán que CloudFormation envíe notificaciones. Cuando se produzcan los eventos especificados por `NotificationTypes`, CloudFormation enviará una notificación al tema de SNS especificado por `TopicARN`. Al lanzar la pila, CloudFormation crea un recurso de [AWS::SNS::Subscription](#) (`snsTopicForAutoScalingGroup`) que se declara dentro de la misma plantilla.

La propiedad `VPCZoneIdentifier` del grupo de escalado automático especifica una lista de subredes existentes en tres zonas de disponibilidad diferentes. Debe especificar los ID de subredes aplicables de su cuenta antes de crear la pila. La propiedad `LaunchTemplate` hace referencia al nombre lógico de un recurso [AWS::EC2::LaunchTemplate](#) declarado en otra parte de la misma plantilla.

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

Creación de un grupo de escalado automático que utiliza una **CreationPolicy** y una **UpdatePolicy**

En el siguiente ejemplo se muestra cómo agregar atributos **CreationPolicy** y **UpdatePolicy** a un recurso de [AWS::AutoScaling::AutoScalingGroup](#).

La política de creación de muestra impide que el grupo de escalado automático alcance el estado **CREATE_COMPLETE** hasta que CloudFormation reciba el número **Count** de señales de éxito cuando el grupo está listo. Para indicar que el grupo de escalado automático está listo, se ejecuta un script del asistente **cfn-signal** agregado a los datos del usuario de la plantilla de lanzamiento (no se muestra) en las instancias. Si las instancias no envían una señal dentro del especificado **Timeout**, CloudFormation asume que las instancias no se crearon, que se produce un error en la creación de recursos y CloudFormation deshace la pila.

La política de actualización de ejemplo indica a CloudFormation que realice una actualización continua con la propiedad **AutoScalingRollingUpdate**. La actualización continua realiza cambios en el grupo de escalado automático en pequeños lotes (para este ejemplo, instancia tras instancia) en función de **MaxBatchSize** y una pausa entre lotes de actualización en función de **PauseTime**. El atributo **MinInstancesInService** especifica el número mínimo de instancias que deben estar en servicio en el grupo de escalado automático mientras CloudFormation actualiza las instancias antiguas.

El atributo **WaitOnResourceSignals** se establece en **true**. CloudFormation debe recibir una señal de cada instancia nueva dentro del valor de **PauseTime** especificado antes de continuar con la actualización. Mientras la actualización de la pila está en curso, se suspenden los siguientes procesos de escalado automático de EC2: **HealthCheck**, **ReplaceUnhealthy**, **AZRebalance**, **AlarmNotification** y **ScheduledActions**. Nota: No suspenda los tipos de proceso **Launch**, **Terminate** ni **AddToLoadBalancer** (si se utiliza el grupo de escalado automático con **Elastic Load Balancing**) ya que ello puede impedir que la actualización continua funcione correctamente.

La propiedad **VPCZoneIdentifier** del grupo de escalado automático especifica una lista de subredes existentes en tres zonas de disponibilidad diferentes. Debe especificar los ID de subredes aplicables de su cuenta antes de crear la pila. La propiedad **LaunchTemplate** hace referencia al

nombre lógico de un recurso [AWS::EC2::LaunchTemplate](#) declarado en otra parte de la misma plantilla.

Para obtener más información sobre los atributos `CreationPolicy` y `UpdatePolicy`, consulte [Referencia de atributos personalizados](#).

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'
```

Creación de una política de escalado por pasos

En este ejemplo, se muestra un recurso de [AWS::AutoScaling::ScalingPolicy](#) que escala horizontalmente el grupo de escalado automático mediante una política de escalado de pasos. La propiedad `AdjustmentType` especifica `ChangeInCapacity`, lo que significa que `ScalingAdjustment` representa el número de instancias que agregar (si `ScalingAdjustment` es positivo) o eliminar (si es negativo). En este ejemplo, `ScalingAdjustment` es 1; por lo tanto, la política aumenta la cantidad de instancias de EC2 del grupo en 1 cuando se infringe el umbral de alarma.

El recurso de [AWS::CloudWatch::Alarm](#) de `CPUAlarmHigh` especifica la política de escalado de `ASGScalingPolicyHigh` como la acción a ejecutar cuando la alarma está en estado de ALARMA (`AlarmActions`). La propiedad `Dimensions` hace referencia al nombre lógico de un recurso [AWS::AutoScaling::AutoScalingGroup](#) declarado en otra parte de la misma plantilla.

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": "GreaterThanOrEqualTo",
    "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
```

Véase también

Para obtener más ejemplos de plantillas para escalar políticas, consulte la sección [Examples](#) (Ejemplos) en el recurso `AWS::AutoScaling::ScalingPolicy`.

Ejemplos del grupo de instancias mixtas

Crear un grupo de escalado automático mediante la selección del tipo de instancia basada en atributos

En este ejemplo, se muestra un recurso de [AWS::AutoScaling::AutoScalingGroup](#) que contiene la información para lanzar un grupo de instancias mixtas mediante la selección del tipo de instancias basadas en atributos. Usted especifica los valores mínimo y máximo para la propiedad `VCpuCount` y el valor mínimo para la propiedad `MemoryMiB`. Todos los tipos de instancia que utiliza el grupo de escalado automático deben coincidir con los atributos de instancia requeridos.

La propiedad `VPCZoneIdentifier` del grupo de escalado automático especifica una lista de subredes existentes en tres zonas de disponibilidad diferentes. Debe especificar los ID de subredes aplicables de su cuenta antes de crear la pila. La propiedad `LaunchTemplate` hace referencia al nombre lógico de un recurso [AWS::EC2::LaunchTemplate](#) declarado en otra parte de la misma plantilla.

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'

```

Ejemplos de configuración de lanzamiento

Creación de una configuración de lanzamiento

En este ejemplo se muestra un recurso de [AWS::AutoScaling::LaunchConfiguration](#) para un grupo de escalado automático en el que especifica valores para las propiedades ImageId, InstanceType y SecurityGroups. La propiedad SecurityGroups especifica el nombre lógico de un recurso de [AWS::EC2::SecurityGroup](#) especificado en otra parte de la plantilla y de un grupo de seguridad de EC2 existente denominado 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
```

Creación de un grupo de escalado automático que utiliza una configuración de lanzamiento

En este ejemplo se muestra un recurso de [AWS::AutoScaling::AutoScalingGroup](#) con una sola instancia. La propiedad `VPCZoneIdentifier` del grupo de escalado automático especifica una lista de subredes existentes en tres zonas de disponibilidad diferentes. Debe especificar los ID de subredes aplicables de su cuenta antes de crear la pila. La propiedad `LaunchConfigurationName` hace referencia a un recurso de [AWS::AutoScaling::LaunchConfiguration](#) con el nombre lógico `mySimpleConfig` definido en la plantilla.

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'
```

Configuración de recursos de escalado automático de aplicaciones con CloudFormation

Esta sección proporciona ejemplos de plantillas de CloudFormation para las políticas de escalado de Application Auto Scaling y acciones programadas para distintos recursos de AWS.

Important

Cuando se incluye un fragmento de escalado automático de aplicaciones en la plantilla, es probable que deba declarar una dependencia del recurso escalable específico que se crea a través de la plantilla mediante el atributo `DependsOn`. Esto anula el paralelismo predeterminado y dirige CloudFormation para operar con recursos en un orden especificado. De lo contrario, la configuración de escalado podría aplicarse antes de que el recurso se haya configurado completamente.

Para obtener más información, consulte [DependsOn atributo](#).

Categorías de fragmentos

- [Creación de una política de escalado para una flota de AppStream](#)
- [Creación de una política de escalado para un clúster de base de datos \(DB\) de Aurora](#)
- [Creación de una política de escalado para una tabla de DynamoDB](#)
- [Creación de una política de escalado para un servicio de Amazon ECS \(métricas: CPU y memoria promedio\)](#)
- [Creación de una política de escalado para un servicio de Amazon ECS \(métrica: recuento de solicitudes promedio por objetivo\)](#)
- [Creación de una acción programada con una expresión cron para una función de Lambda](#)
- [Creación de una acción programada con una expresión at para una flota de spot](#)

Creación de una política de escalado para una flota de AppStream

Este fragmento muestra cómo crear una política y aplicarla a recurso de [AWS::AppStream::Fleet](#) por medio del recurso [AWS::ApplicationAutoScaling::ScalingPolicy](#). El recurso [AWS::ApplicationAutoScaling::ScalableTarget](#) declara un destino escalable al que se aplica esta política. Application Auto Scaling puede escalar el número de instancias de flota en un mínimo de 1 instancia y un máximo de 20. La política mantiene el uso promedio de la capacidad

de la flota en un 75 %, con periodos de recuperación de escalamiento y reducción horizontal de 300 segundos (5 minutos).

Utiliza las funciones intrínsecas `Fn::Join` y `Rev` para construir la propiedad `ResourceId` con el nombre lógico del recurso de `AWS::AppStream::Fleet` especificado en la misma plantilla. Para obtener más información, consulte [Referencia de función intrínseca](#).

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

```

Creación de una política de escalado para un clúster de base de datos (DB) de Aurora

En este fragmento, registra un recurso de [AWS::RDS::DBCluster](#). El recurso de [AWS::ApplicationAutoScaling::ScalableTarget](#) indica que el clúster de base de datos debe escalarse en forma dinámica para tener de una a ocho réplicas de Aurora. También aplica una política de escalado de seguimiento de destino al clúster mediante el recurso de [AWS::ApplicationAutoScaling::ScalingPolicy](#).

En esta configuración, la métrica predefinida `RDSReaderAverageCPUUtilization` se usa para ajustar un clúster de base de datos de Aurora en función del uso promedio de la CPU del 40 por ciento en todas las réplicas de Aurora de ese clúster de base de datos de Aurora. La configuración proporciona un periodo de recuperación de escalado descendente de 10 minutos y un periodo de recuperación de escalado ascendente de 5 minutos.

En este ejemplo, se usa la función intrínseca `Fn::Sub` para construir la propiedad `ResourceId` con el nombre lógico del recurso de `AWS::RDS::DBCluster` especificado en la misma plantilla. Para obtener más información, consulte [Referencia de función intrínseca](#).

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
```

Creación de una política de escalado para una tabla de DynamoDB

Este fragmento muestra cómo crear un tipo de política de `TargetTrackingScaling` y aplicarla a un recurso de [AWS::DynamoDB::Table](#) por medio del recurso de [AWS::ApplicationAutoScaling::ScalingPolicy](#). El recurso de [AWS::ApplicationAutoScaling::ScalableTarget](#) declara un destino escalable al que se aplica esta política, con un mínimo de cinco unidades de capacidad de escritura y un máximo de 15. La política de escalado escala el rendimiento de la capacidad de escritura de la tabla para mantener la utilización de destino en un 50 por ciento en función de la métrica predefinida `DynamoDBWriteCapacityUtilization`.

Utiliza las funciones intrínsecas `Fn::Join` y `Ref` para construir la propiedad `ResourceId` con el nombre lógico del recurso de `AWS::DynamoDB::Table` especificado en la misma plantilla. Para obtener más información, consulte [Referencia de función intrínseca](#).

Note

Para obtener más información acerca de cómo crear una plantilla de CloudFormation para DynamoDB, consulte la publicación del blog [Cómo utilizar CloudFormation para configurar el escalado automático para las tablas e índices de Amazon DynamoDB](#) en el blog de la base de datos de 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
```

Creación de una política de escalado para un servicio de Amazon ECS (métricas: CPU y memoria promedio)

Este fragmento muestra cómo crear una política y aplicarla a un recurso de [AWS::ECS::Service](#) por medio del recurso [AWS::ApplicationAutoScaling::ScalingPolicy](#). El recurso de [AWS::ApplicationAutoScaling::ScalableTarget](#) declara un destino escalable al que se aplica esta política. El escalado automático de aplicaciones puede escalar el número de tareas a un mínimo de 1 tarea y un máximo de 6.

Crea dos políticas de escalado con el tipo de política `TargetTrackingScaling`. Las políticas se utilizan para escalar el servicio ECS en función del uso medio de CPU y memoria del servicio. Utiliza las funciones intrínsecas `Fn::Join` y `Ref` para crear la propiedad `ResourceId` con los nombres lógicos de los recursos de [AWS::ECS::Cluster](#) (`myContainerCluster`) y `AWS::ECS::Service` (`myService`) especificados en la misma plantilla. Para obtener más información, consulte [Referencia de función intrínseca](#).

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

```

Creación de una política de escalado para un servicio de Amazon ECS (métrica: recuento de solicitudes promedio por objetivo)

En el ejemplo siguiente se aplica una política de escalado de seguimiento de destino con la métrica predefinida `ALBRequestCountPerTarget` a un servicio ECS. La política se utiliza para agregar capacidad al servicio de ECS cuando el recuento de solicitudes por destino (por minuto) supera el valor de destino. Dado que el valor de `DisableScaleIn` se establece en `true`, la política de seguimiento de destino no eliminará capacidad del destino escalable.

Utiliza las funciones intrínsecas `Fn::Join` y `Fn::GetAtt` para crear la propiedad `ResourceLabel` con los nombres lógicos de los recursos de [AWS::ElasticLoadBalancingV2::LoadBalancer](#) (`myLoadBalancer`) y [AWS::ElasticLoadBalancingV2::TargetGroup](#) (`myTargetGroup`) especificados en la misma plantilla. Para obtener más información, consulte [Referencia de función intrínseca](#).

Las propiedades `MaxCapacity` y `MinCapacity` del destino escalable y la propiedad `TargetValue` de la política de escalado hacen referencia a los valores de parámetro que se pasan a la plantilla al crear o actualizar una pila.

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.TargetGroupName

```

Creación de una acción programada con una expresión cron para una función de Lambda

Este fragmento registra la simultaneidad aprovisionada para un alias de función ([AWS::Lambda::Alias](#)) denominado BLUE mediante el recurso de [AWS::ApplicationAutoScaling::ScalableTarget](#). También crea una acción programada con una programación periódica utilizando una expresión cron. La zona horaria del programa recurrente es UTC.

Utiliza las funciones intrínsecas `Fn::Join` y `Ref` de la propiedad `RoleARN` para especificar el ARN del rol vinculado al servicio. Utiliza la función intrínseca `Fn::Sub` para construir la

propiedad `ResourceId` con el nombre lógico del recurso de [AWS::Lambda::Function](#) o [AWS::Serverless::Function](#) que se especifica en la misma plantilla. Para obtener más información, consulte [Referencia de función intrínseca](#).

Note

Tampoco puede asignar simultaneidad aprovisionada en un alias que apunte a la versión no publicada (\$LATEST).

Para obtener más información acerca de cómo crear una plantilla de CloudFormation para recursos de Lambda, consulte la publicación del blog [Programación de la simultaneidad aprovisionada de AWS Lambda durante el uso en horarios de actividad alta simultánea](#) en el blog de computación de 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'

```

Creación de una acción programada con una expresión **at** para una flota de spot

Este fragmento muestra cómo crear dos acciones programadas que se produzcan solo una vez para un recurso de [AWS::EC2::SpotFleet](#) que utilice el recurso de [AWS::ApplicationAutoScaling::ScalableTarget](#). La zona horaria de cada acción programada puntual es UTC.

Utiliza las funciones intrínsecas `Fn::Join` y `Ref` para construir la propiedad `ResourceId` con el nombre lógico del recurso de `AWS::EC2::SpotFleet` especificado en la misma plantilla. Para obtener más información, consulte [Referencia de función intrínseca](#).

Note

La solicitud de flota de spot debe tener un tipo de solicitud de `maintain`. El escalado automático no se admite para solicitudes o bloques de spot de una única vez.

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)'
```

Fragmentos de plantillas de la consola de facturación de AWS

En este ejemplo, se crea un plan de precios con una regla de precios con un margen global del 10 %. Este plan de precios se adjunta al grupo de facturación. El grupo de facturación también tiene dos líneas de pedido personalizadas que aplican un cargo de 10 USD y un cargo del 10 % sobre el costo total del grupo de facturación.

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'

```

CloudFormationFragmentos de plantillas de

Temas

- [Pilas anidadas](#)
- [Condición de espera](#)

Pilas anidadas

Anidación de una pila en una plantilla

Esta plantilla de ejemplo contiene un recurso de pilas anidadas denominadas `myStack`. Cuando CloudFormation crea una pila a partir de la plantilla, crea `myStack`, cuya plantilla se especifica en la propiedad `TemplateURL`. El valor de salida `StackRef` devuelve el ID de pila `myStack` y el valor `OutputFromNestedStack` devuelve el valor de salida `BucketName` desde el recurso `myStack`. El formato `Outputs.nestedstackoutputname` se reserva para especificar valores de salida de pilas anidadas y se puede utilizar en cualquier parte de la plantilla contenedora.

Para obtener más información, consulte [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
```

Anidar una pila con parámetros de entrada en una plantilla

Esta plantilla de ejemplo contiene un recurso de un conjunto que especifica parámetros de entrada. Cuando CloudFormation crea una pila a partir de esta plantilla, utiliza los pares de valores declarados dentro de la propiedad `Parameters` como parámetros de entrada para la plantilla utilizada para crear la pila `myStackWithParams`. En este ejemplo, se especifican los parámetros `InstanceType` y `KeyName`.

Para obtener más información, consulte [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
```

Condición de espera

Uso de una condición de espera con una instancia Amazon EC2

Important

Con los recursos de Amazon EC2 y Auto Scaling recomendamos que use un atributo `CreationPolicy` en lugar de condiciones de espera. Agregue un atributo `CreationPolicy` a esos recursos y use el script de ayuda `cfn-signal` para señalar que el proceso de creación de instancias ha finalizado satisfactoriamente.

Si no puede utilizar una política de creación, ve la siguiente plantilla de ejemplo, que declara una instancia Amazon EC2 con una condición de espera. La condición de espera `myWaitCondition` utiliza `myWaitConditionHandle` para señalización, utiliza el atributo `DependsOn` para especificar que la condición de espera se activará después de que se haya creado el recurso de la instancia de Amazon EC2 y utiliza la propiedad `Timeout` para especificar una duración de 4500 segundos para la condición de espera. Además, la URL prefirmada que señala la condición de espera se transfiere a la instancia de Amazon EC2 con la propiedad `UserData` del recurso de `Ec2Instance`, lo que permite que se ejecute una aplicación o script en dicha instancia de Amazon EC2 para recuperar la URL prefirmada y emplearla para señalar un éxito o error a la condición de espera. Necesita usar `cfn-signal` o crear la aplicación o script que señale la condición de espera. El valor de salida `ApplicationData` contiene los datos devueltos desde la señal de condición de espera.

Para obtener más información, consulte [Creación de condiciones de espera en una plantilla de CloudFormation](#).

JSON

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Mappings" : {
```

```

    "RegionMap" : {
      "us-east-1" : {
        "AMI" : "ami-0123456789abcdef0"
      },
      "us-west-1" : {
        "AMI" : "ami-0987654321fedcba0"
      },
      "eu-west-1" : {
        "AMI" : "ami-0abcdef123456789a"
      },
      "ap-northeast-1" : {
        "AMI" : "ami-0fedcba987654321b"
      },
      "ap-southeast-1" : {
        "AMI" : "ami-0c1d2e3f4a5b6c7d8"
      }
    },
    "Resources" : {
      "Ec2Instance" : {
        "Type" : "AWS::EC2::Instance",
        "Properties" : {
          "UserData" : { "Fn::Base64" : { "Ref" : "myWaitHandle" } },
          "ImageId" : { "Fn::FindInMap" : [ "RegionMap", { "Ref" :
"AWS::Region" }, "AMI" ] }
        }
      },
      "myWaitHandle" : {
        "Type" : "AWS::CloudFormation::WaitConditionHandle",
        "Properties" : {
        }
      },
      "myWaitCondition" : {
        "Type" : "AWS::CloudFormation::WaitCondition",
        "DependsOn" : "Ec2Instance",
        "Properties" : {
          "Handle" : { "Ref" : "myWaitHandle" },
          "Timeout" : "4500"
        }
      }
    },
    "Outputs" : {
      "ApplicationData" : {
        "Value" : { "Fn::GetAtt" : [ "myWaitCondition", "Data" ] },

```

```

        "Description" : "The data passed back as part of signalling the
WaitCondition."
    }
}
}

```

YAML

```

AWSTemplateFormatVersion: '2010-09-09'
Mappings:
  RegionMap:
    us-east-1:
      AMI: ami-0123456789abcdef0
    us-west-1:
      AMI: ami-0987654321fedcba0
    eu-west-1:
      AMI: ami-0abcdef123456789a
    ap-northeast-1:
      AMI: ami-0fedcba987654321b
    ap-southeast-1:
      AMI: ami-0c1d2e3f4a5b6c7d8
Resources:
  Ec2Instance:
    Type: AWS::EC2::Instance
    Properties:
      UserData:
        Fn::Base64: !Ref myWaitHandle
      ImageId:
        Fn::FindInMap:
          - RegionMap
          - Ref: AWS::Region
          - AMI
  myWaitHandle:
    Type: AWS::CloudFormation::WaitConditionHandle
    Properties: {}
  myWaitCondition:
    Type: AWS::CloudFormation::WaitCondition
    DependsOn: Ec2Instance
    Properties:
      Handle: !Ref myWaitHandle
      Timeout: '4500'
Outputs:
  ApplicationData:

```

Value: !GetAtt myWaitCondition.Data

Description: The data passed back as part of signalling the WaitCondition.

Uso del script auxiliar cfn-signal para señalar a una condición de espera

En este ejemplo se muestra una línea de comandos `cfn-signal` que indica el éxito a una condición de espera. Tiene que definir la línea de comandos en la propiedad `UserData` de la instancia de EC2.

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
          - |
            ,
```

Uso de Curl para señalar una condición de espera

En este ejemplo se muestra una línea de comandos Curl que indica el éxito a una condición de espera.

```
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"
```

Donde el archivo /tmp/a contiene la siguiente estructura JSON:

```
{
  "Status" : "SUCCESS",
  "Reason" : "Configuration Complete",
  "UniqueId" : "ID1234",
  "Data" : "Application has completed configuration."
}
```

En este ejemplo se muestra una línea de comandos Curl que envía la misma señal de éxito, salvo que envía la señal JSON como parámetro en la línea de comandos.

```
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"
```

Fragmentos de código de plantillas de Amazon CloudFront

Utilice estos fragmentos de código de plantillas de ejemplo con su recurso de distribución de Amazon CloudFront en CloudFormation. Para obtener más información, consulte [Amazon CloudFront resource type reference](#).

Temas

- [Recurso de distribución de Amazon CloudFront con origen Amazon S3](#)

- [Recurso de distribución de Amazon CloudFront con origen personalizado](#)
- [Distribución de Amazon CloudFront con soporte de origen múltiple](#)
- [Distribución de Amazon CloudFront con una función de Lambda como origen](#)
- [Véase también](#)

Recurso de distribución de Amazon CloudFront con origen Amazon S3

En la siguiente plantilla de ejemplo se muestra una [distribución](#) de Amazon CloudFront mediante un identidad de acceso de origen (OAI) heredada y [S3Origin](#). Para obtener información acerca de cómo utilizar un control de acceso de origen (OAC) en su lugar, consulte [Restricción del acceso a un origen de Amazon Simple Storage Service](#) en la Guía para desarrolladores de Amazon CloudFront.

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'
```

Recurso de distribución de Amazon CloudFront con origen personalizado

La siguiente plantilla de ejemplo muestra una [Distribución](#) de Amazon CloudFront mediante un [CustomOrigin](#).

JSON

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Resources" : {
    "myDistribution" : {
      "Type" : "AWS::CloudFront::Distribution",
      "Properties" : {
```

```
"DistributionConfig" : {
  "Origins" : [ {
    "DomainName" : "www.example.com",
    "Id" : "myCustomOrigin",
    "CustomOriginConfig" : {
      "HTTPPort" : "80",
      "HTTPSPort" : "443",
      "OriginProtocolPolicy" : "http-only"
    }
  } ],
  "Enabled" : "true",
  "Comment" : "Somecomment",
  "DefaultRootObject" : "index.html",
  "Logging" : {
    "IncludeCookies" : "true",
    "Bucket" : "amzn-s3-demo-logging-bucket.s3.amazonaws.com",
    "Prefix" : "myprefix"
  },
  "Aliases" : [
    "mysite.example.com",
    "*.yoursite.example.com"
  ],
  "DefaultCacheBehavior" : {
    "TargetOriginId" : "myCustomOrigin",
    "SmoothStreaming" : "false",
    "ForwardedValues" : {
      "QueryString" : "false",
      "Cookies" : { "Forward" : "all" }
    },
    "TrustedSigners" : [
      "1234567890EX",
      "1234567891EX"
    ],
    "ViewerProtocolPolicy" : "allow-all"
  },
  "CustomErrorResponses" : [ {
    "ErrorCode" : "404",
    "ResponsePagePath" : "/error-pages/404.html",
    "ResponseCode" : "200",
    "ErrorCachingMinTTL" : "30"
  } ],
  "PriceClass" : "PriceClass_200",
  "Restrictions" : {
    "GeoRestriction" : {
```

```
    "RestrictionType" : "whitelist",
    "Locations" : [ "AQ", "CV" ]
  },
  "ViewerCertificate": { "CloudFrontDefaultCertificate" : "true" }
}
}
```

YAML

```

AWSTemplateFormatVersion: '2010-09-09'
Resources:
  myDistribution:
    Type: AWS::CloudFront::Distribution
    Properties:
      DistributionConfig:
        Origins:
          - DomainName: www.example.com
            Id: myCustomOrigin
            CustomOriginConfig:
              HTTPPort: '80'
              HTTPSPort: '443'
              OriginProtocolPolicy: http-only
        Enabled: 'true'
        Comment: Somecomment
        DefaultRootObject: index.html
        Logging:
          IncludeCookies: 'true'
          Bucket: amzn-s3-demo-logging-bucket.s3.amazonaws.com
          Prefix: myprefix
        Aliases:
          - mysite.example.com
          - "*.yoursite.example.com"
        DefaultCacheBehavior:
          TargetOriginId: myCustomOrigin
          SmoothStreaming: 'false'
          ForwardedValues:
            QueryString: 'false'
            Cookies:
              Forward: all

```

```

TrustedSigners:
- 1234567890EX
- 1234567891EX
ViewerProtocolPolicy: allow-all
CustomErrorResponses:
- ErrorCode: '404'
  ResponsePagePath: "/error-pages/404.html"
  ResponseCode: '200'
  ErrorCachingMinTTL: '30'
PriceClass: PriceClass_200
Restrictions:
  GeoRestriction:
    RestrictionType: whitelist
    Locations:
      - AQ
      - CV
ViewerCertificate:
  CloudFrontDefaultCertificate: 'true'

```

Distribución de Amazon CloudFront con soporte de origen múltiple

La siguiente plantilla de ejemplo muestra cómo declarar una [Distribución](#) de CloudFront con soporte de origen múltiple. En la [DistributionConfig](#), se proporciona una lista de orígenes y se establece un [DefaultCacheBehavior](#).

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

```
AWS::CloudFront::Distribution
```

```
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'
```

Distribución de Amazon CloudFront con una función de Lambda como origen

En el siguiente ejemplo se crea una distribución de CloudFront que sirve de frente para la URL de una función de Lambda especificada (que se proporciona como parámetro), lo que permite el acceso solo mediante HTTPS, el almacenamiento en caché, la compresión y la entrega global. Configura la URL de Lambda como un origen HTTPS personalizado y aplica una política de almacenamiento en caché de AWS estándar. La distribución está optimizada para el rendimiento con compatibilidad con HTTP/2 e IPv6 y muestra el nombre de dominio de CloudFront, lo que permite a los usuarios acceder a la función de Lambda a través de un punto de conexión seguro respaldado por una CDN. Para obtener más información, consulte [Using Amazon CloudFront with AWS Lambda as origin to accelerate your web applications](#) en el blog de AWS.

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}/

```

Véase también

Para ver un ejemplo de cómo agregar un alias personalizado a un registro de Route 53 para crear un nombre descriptivo para una distribución de CloudFront, consulte [Conjunto de registros de recursos de alias para una distribución de CloudFront](#).

Fragmentos de código de plantillas de Amazon CloudWatch

Use estos ejemplos de fragmentos de código de plantillas para describir los recursos de Amazon CloudWatch en CloudFormation. Para obtener más información, consulte la [Referencia del tipo de recurso de Amazon CloudWatch](#).

Temas

- [Alarma de facturación](#)
- [Alarma de utilización de la CPU](#)
- [Recuperación de una instancia Amazon Elastic Compute Cloud](#)
- [Creación de un panel básico](#)
- [Creación de un panel con widgets en paralelo](#)

Alarma de facturación

En el siguiente ejemplo, Amazon CloudWatch envía una notificación por correo electrónico cuando los cargos de una cuenta de AWS superan el umbral de alarma. Para recibir notificaciones de uso, habilite las alertas de facturación. Para obtener más información, consulte [Crear una alarma de facturación para supervisar los cargos estimados de AWS](#) en la Guía del usuario de Amazon CloudWatch.>

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": "GreaterThanOrEqualToThreshold",
    "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: GreaterThanOrEqualToThreshold
    AlarmActions:
      - !Ref: "BillingAlarmNotification"
    InsufficientDataActions:
      - !Ref: "BillingAlarmNotification"

```

Alarma de utilización de la CPU

El siguiente fragmento de código de ejemplo crea una alarma que envía una notificación cuando el uso promedio de la CPU de una instancia Amazon EC2 supera el 90 por ciento durante más de 60 segundos durante tres periodos de evaluación.

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*"

Recuperación de una instancia Amazon Elastic Compute Cloud

La siguiente alarma de CloudWatch recupera una instancia de EC2 cuando presenta errores de comprobación de estado durante 15 minutos seguidos. Para obtener más información acerca de las acciones de alarma, consulte [Crear alarmas para detener, terminar, reiniciar o recuperar una instancia de EC2](#) en la Guía del usuario de Amazon CloudWatch.

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

```

Creación de un panel básico

El siguiente ejemplo crea un panel de CloudWatch sencillo con un widget de métrica que muestra la utilización de CPU, y un widget de texto que muestra un mensaje.

JSON

```

{
  "BasicDashboard": {
    "Type": "AWS::CloudWatch::Dashboard",
    "Properties": {
      "DashboardName": "Dashboard1",
      "DashboardBody": "{\n  \"widgets\": [\n    {\n      \"type\": \"metric\",\n      \"x\": 0,\n      \"y\": 0,\n      \"width\": 12,\n      \"height\": 6,\n      \"properties\": {\n        \"metrics\": [\n          [\n            \"AWS/EC2\",\n            \"CPUUtilization\",\n            \"InstanceId\",\n            \"i-012345\"\n          ],\n          {\n            \"period\": 300,\n            \"stat\": \"Average\",\n            \"region\": \"us-east-1\",\n            \"title\": \"EC2 Instance CPU\"\n          }\n        ]\n      },\n      {\n        \"type\": \"text\",\n        \"x\": 0,\n        \"y\": 7,\n        \"width\": 3,\n        \"height\": 3,\n        \"properties\": {\n          \"markdown\": \"Hello world\"\n        }\n      }\n    ]\n  }\n}"
    }
  }
}

```



```

    }
  }
}

```

YAML

```

BasicDashboard:
  Type: AWS::CloudWatch::Dashboard
  Properties:
    DashboardName: Dashboard1
    DashboardBody: '{"widgets":
[{"type":"metric","x":0,"y":0,"width":12,"height":6,"properties":{"metrics":[["AWS/
EC2","CPUUtilization","InstanceId","i-012345"]], "period":300,"stat":"Average","region":"us-
east-1","title":"EC2 Instance CPU"}},
{"type":"text","x":0,"y":7,"width":3,"height":3,"properties":{"markdown":"Hello
world"}}]}'

```

Creación de un panel con widgets en paralelo

El siguiente ejemplo crea un panel con dos widgets de métricas que aparecen uno al lado del otro.

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"}}]']
```

Fragmentos de la plantilla de Amazon CloudWatch Logs

Amazon CloudWatch Logs puede monitorizar sus archivos de registro personalizados, de las aplicaciones y del sistema desde instancias Amazon EC2 u otros orígenes. Puede usar CloudFormation para aprovisionar y administrar grupos de registro y filtros de métricas. Para obtener más información acerca de Registros de Amazon CloudWatch, consulte la [Guía del usuario de Registros de Amazon CloudWatch](#).

Temas

- [Enviar registros a CloudWatch Logs desde una instancia Linux](#)
- [Enviar registros a CloudWatch Logs desde una instancia Windows](#)
- [Véase también](#)

Enviar registros a CloudWatch Logs desde una instancia Linux

La siguiente plantilla muestra cómo configurar un servidor web en Amazon Linux 2023 con la integración de Registros de CloudWatch. La plantilla realiza las siguientes tareas:

- Instala Apache y PHP.
- Configura el agente de CloudWatch para reenviar los registros de acceso de Apache a Registros de CloudWatch.
- Configura un rol de IAM para permitir al agente de CloudWatch enviar datos de registro a Registros de CloudWatch.
- Crea alarmas y notificaciones personalizadas para detectar errores 404 o un uso elevado del ancho de banda.

Los eventos de registro del servidor web proporcionan datos de métricas para las alarmas de CloudWatch. Los dos filtros de métrica describen cómo la información de registro se transforma en métricas de CloudWatch. La métrica 404 cuenta el número de casos de 404. La métrica de tamaño métrica controla el tamaño de una solicitud. Las dos alarmas de CloudWatch enviarán notificaciones si hay más de dos 404 en un plazo de 2 minutos o si el tamaño de solicitud media es de más de 3500 KB en 10 minutos.

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "Sample template that sets up and configures CloudWatch Logs on Amazon Linux 2023 instance.",
  "Parameters": {
    "KeyName": {
      "Description": "Name of an existing EC2 KeyPair to enable SSH access to the instances",
      "Type": "AWS::EC2::KeyPair::KeyName",
      "ConstraintDescription": "must be the name of an existing EC2 KeyPair."
    },
    "SSHLocation": {
      "Description": "The IP address range that can be used to SSH to the EC2 instances",
      "Type": "String",
      "MinLength": "9",
      "MaxLength": "18",
      "Default": "0.0.0.0/0",
      "AllowedPattern": "(\\d{1,3})\\. (\\d{1,3})\\. (\\d{1,3})\\. (\\d{1,3})/(\\d{1,2})",
      "ConstraintDescription": "must be a valid IP CIDR range of the form x.x.x.x/x."
    },
    "OperatorEmail": {
      "Description": "Email address to notify when CloudWatch alarms are triggered (404 errors or high bandwidth usage)",
      "Type": "String"
    }
  },
  "Resources": {
    "LogRole": {
      "Type": "AWS::IAM::Role",
      "Properties": {
        "AssumeRolePolicyDocument": {
```

```

        "Version": "2012-10-17",
        "Statement": [
            {
                "Effect": "Allow",
                "Principal": {
                    "Service": [
                        "ec2.amazonaws.com"
                    ]
                },
                "Action": [
                    "sts:AssumeRole"
                ]
            }
        ],
        "Path": "/",
        "Policies": [
            {
                "PolicyName": "LogRolePolicy",
                "PolicyDocument": {
                    "Version": "2012-10-17",
                    "Statement": [
                        {
                            "Effect": "Allow",
                            "Action": [
                                "logs:PutLogEvents",
                                "logs:DescribeLogStreams",
                                "logs:DescribeLogGroups",
                                "logs:CreateLogGroup",
                                "logs:CreateLogStream"
                            ],
                            "Resource": "*"
                        }
                    ]
                }
            }
        ]
    },
    "LogRoleInstanceProfile": {
        "Type": "AWS::IAM::InstanceProfile",
        "Properties": {
            "Path": "/",
            "Roles": [{"Ref": "LogRole"}]
        }
    }
}

```

```

    }
  },
  "WebServerSecurityGroup": {
    "Type": "AWS::EC2::SecurityGroup",
    "Properties": {
      "GroupDescription": "Enable HTTP access via port 80 and SSH access via
port 22",
      "SecurityGroupIngress": [
        {
          "IpProtocol": "tcp",
          "FromPort": 80,
          "ToPort": 80,
          "CidrIp": "0.0.0.0/0"
        },
        {
          "IpProtocol": "tcp",
          "FromPort": 22,
          "ToPort": 22,
          "CidrIp": {"Ref": "SSHLocation"}
        }
      ]
    }
  },
  "WebServerHost": {
    "Type": "AWS::EC2::Instance",
    "Metadata": {
      "Comment": "Install a simple PHP application on Amazon Linux 2023",
      "AWS::CloudFormation::Init": {
        "config": {
          "packages": {
            "dnf": {
              "httpd": [],
              "php": [],
              "php-fpm": []
            }
          },
          "files": {
            "/etc/amazon-cloudwatch-agent/amazon-cloudwatch-
agent.json": {
              "content": {
                "logs": {
                  "logs_collected": {
                    "files": {
                      "collect_list": [{

```

```

        "file_path": "/var/log/httpd/
access_log",
        "log_group_name": {"Ref":
        "WebServerLogGroup"},
        "log_stream_name": "{instance_id}/
apache.log",
        "timestamp_format": "%d/%b/%Y:%H:
%M:%S %z"
    }
}
}
},
{
    "mode": "000644",
    "owner": "root",
    "group": "root"
},
"/var/www/html/index.php": {
    "content": "<?php\\necho '<h1>AWS CloudFormation sample
PHP application on Amazon Linux 2023</h1>';\\n?>\\n",
    "mode": "000644",
    "owner": "apache",
    "group": "apache"
},
"/etc/cfn/cfn-hup.conf": {
    "content": {
        "Fn::Join": [
            "",
            [
                "[main]\\n",
                "stack=",
                {"Ref": "AWS::StackId"},
                "\\n",
                "region=",
                {"Ref": "AWS::Region"},
                "\\n"
            ]
        ]
    },
    "mode": "000400",
    "owner": "root",
    "group": "root"
},
"/etc/cfn/hooks.d/cfn-auto-reloader.conf": {

```

```

        "content": {
            "Fn::Join": [
                "",
                [
                    "[cfn-auto-reloader-hook]\n",
                    "triggers=post.update\n",
                    "path=Resources.WebServerHost.Metadata.AWS::CloudFormation::Init\n",
                    "action=/opt/aws/bin/cfn-init -s ",
                    {"Ref": "AWS::StackId"},
                    " -r WebServerHost ",
                    " --region ",
                    {"Ref": "AWS::Region"},
                    "\n",
                    "runas=root\n"
                ]
            ]
        },
        "services": {
            "systemd": {
                "httpd": {
                    "enabled": "true",
                    "ensureRunning": "true"
                },
                "php-fpm": {
                    "enabled": "true",
                    "ensureRunning": "true"
                }
            }
        },
        "CreationPolicy": {
            "ResourceSignal": {
                "Timeout": "PT5M"
            }
        },
        "Properties": {
            "ImageId": "{{resolve:ssm:/aws/service/ami-amazon-linux-latest/al2023-ami-kernel-6.1-x86_64}}",
            "KeyName": {"Ref": "KeyName"},

```

```

      "InstanceType": "t3.micro",
      "SecurityGroupIds": [{"Ref": "WebServerSecurityGroup"}],
      "IamInstanceProfile": {"Ref": "LogRoleInstanceProfile"},
      "UserData": {"Fn::Base64": {"Fn::Join": [ "", [
        "#!/bin/bash\n",
        "dnf update -y aws-cfn-bootstrap\n",
        "dnf install -y amazon-cloudwatch-agent\n",
        "/opt/aws/bin/cfn-init -v --stack ", {"Ref": "AWS::StackName"}, "
--resource WebServerHost --region ", {"Ref": "AWS::Region"}, "\n",
        "\n",
        "# Verify Apache log directory exists and create if needed\n",
        "mkdir -p /var/log/httpd\n",
        "\n",
        "# Start CloudWatch agent\n",
        "/opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -
a fetch-config -m ec2 -c file:/etc/amazon-cloudwatch-agent/amazon-cloudwatch-agent.json
-s\n",
        "\n",
        "# Signal success\n",
        "/opt/aws/bin/cfn-signal -e $? --stack ", {"Ref":
"AWS::StackName"}, " --resource WebServerHost --region ", {"Ref": "AWS::Region"}, "\n"
]]}]
    }
  },
  "WebServerLogGroup": {
    "Type": "AWS::Logs::LogGroup",
    "DeletionPolicy": "Retain",
    "UpdateReplacePolicy": "Retain",
    "Properties": {
      "RetentionInDays": 7
    }
  },
  "404MetricFilter": {
    "Type": "AWS::Logs::MetricFilter",
    "Properties": {
      "LogGroupName": {"Ref": "WebServerLogGroup"},
      "FilterPattern": "[ip, identity, user_id, timestamp, request,
status_code = 404, size, ...]",
      "MetricTransformations": [
        {
          "MetricValue": "1",
          "MetricNamespace": "test/404s",
          "MetricName": "test404Count"
        }
      ]
    }
  }
}

```



```

        ]
    },
    "BytesTransferredMetricFilter": {
        "Type": "AWS::Logs::MetricFilter",
        "Properties": {
            "LogGroupName": {"Ref": "WebServerLogGroup"},
            "FilterPattern": "[ip, identity, user_id, timestamp, request,
status_code, size, ...]",
            "MetricTransformations": [
                {
                    "MetricValue": "$size",
                    "MetricNamespace": "test/BytesTransferred",
                    "MetricName": "testBytesTransferred"
                }
            ]
        }
    },
    "404Alarm": {
        "Type": "AWS::CloudWatch::Alarm",
        "Properties": {
            "AlarmDescription": "The number of 404s is greater than 2 over 2
minutes",
            "MetricName": "test404Count",
            "Namespace": "test/404s",
            "Statistic": "Sum",
            "Period": "60",
            "EvaluationPeriods": "2",
            "Threshold": "2",
            "AlarmActions": [{"Ref": "AlarmNotificationTopic"}],
            "ComparisonOperator": "GreaterThanThreshold"
        }
    },
    "BandwidthAlarm": {
        "Type": "AWS::CloudWatch::Alarm",
        "Properties": {
            "AlarmDescription": "The average volume of traffic is greater 3500 KB
over 10 minutes",
            "MetricName": "testBytesTransferred",
            "Namespace": "test/BytesTransferred",
            "Statistic": "Average",
            "Period": "300",
            "EvaluationPeriods": "2",
            "Threshold": "3500",

```

```

        "AlarmActions": [{"Ref": "AlarmNotificationTopic"}],
        "ComparisonOperator": "GreaterThanThreshold"
    },
    "AlarmNotificationTopic": {
        "Type": "AWS::SNS::Topic",
        "Properties": {
            "Subscription": [{"Endpoint": {"Ref": "OperatorEmail"}, "Protocol":
"email"}]
        }
    },
    "Outputs": {
        "InstanceId": {
            "Description": "The instance ID of the web server",
            "Value": {"Ref": "WebServerHost"}
        },
        "WebsiteURL": {
            "Value": {"Fn::Sub": "http://${WebServerHost.PublicDnsName}"},
            "Description": "URL for the web server"
        },
        "PublicIP": {
            "Description": "Public IP address of the web server",
            "Value": {"Fn::GetAtt": ["WebServerHost", "PublicIp"]}
        },
        "CloudWatchLogGroupName": {
            "Description": "The name of the CloudWatch log group",
            "Value": {"Ref": "WebServerLogGroup"}
        }
    }
}

```

YAML

```

AWSTemplateFormatVersion: 2010-09-09
Description: Sample template that sets up and configures CloudWatch Logs on Amazon
Linux 2023 instance.
Parameters:
  KeyName:
    Description: Name of an existing EC2 KeyPair to enable SSH access to the instances
    Type: AWS::EC2::KeyPair::KeyName
    ConstraintDescription: must be the name of an existing EC2 KeyPair.

```

SSHLocation:

Description: The IP address range that can be used to SSH to the EC2 instances

Type: String

MinLength: '9'

MaxLength: '18'

Default: 0.0.0.0/0

AllowedPattern: '(\d{1,3})\.\d{1,3})\.\d{1,3})\.\d{1,3})/(\d{1,2})'

ConstraintDescription: must be a valid IP CIDR range of the form x.x.x.x/x.

OperatorEmail:

Description: Email address to notify when CloudWatch alarms are triggered (404 errors or high bandwidth usage)

Type: String

Resources:**LogRole:**

Type: AWS::IAM::Role

Properties:

AssumeRolePolicyDocument:

Version: 2012-10-17

Statement:

- Effect: Allow

Principal:

Service:

- ec2.amazonaws.com

Action:

- 'sts:AssumeRole'

Path: /

Policies:

- PolicyName: LogRolePolicy

PolicyDocument:

Version: 2012-10-17

Statement:

- Effect: Allow

Action:

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

Resource: '*'

LogRoleInstanceProfile:

Type: AWS::IAM::InstanceProfile

Properties:

Path: /

Roles:

```

    - !Ref LogRole
WebServerSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Enable HTTP access via port 80 and SSH access via port 22
    SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: 80
        ToPort: 80
        CidrIp: 0.0.0.0/0
      - IpProtocol: tcp
        FromPort: 22
        ToPort: 22
        CidrIp: !Ref SSHLocation
WebServerHost:
  Type: AWS::EC2::Instance
  Metadata:
    Comment: Install a simple PHP application on Amazon Linux 2023
    'AWS::CloudFormation::Init':
      config:
        packages:
          dnf:
            httpd: []
            php: []
            php-fpm: []
        files:
          /etc/amazon-cloudwatch-agent/amazon-cloudwatch-agent.json:
            content: !Sub |
              {
                "logs": {
                  "logs_collected": {
                    "files": {
                      "collect_list": [
                        {
                          "file_path": "/var/log/httpd/access_log",
                          "log_group_name": "${WebServerLogGroup}",
                          "log_stream_name": "{instance_id}/apache.log",
                          "timestamp_format": "%d/%b/%Y:%H:%M:%S %z"
                        }
                      ]
                    }
                  }
                }
              }

```

```

        mode: '000644'
        owner: root
        group: root
/var/www/html/index.php:
    content: |
        <?php echo '<h1>AWS CloudFormation sample PHP application on Amazon
Linux 2023</h1>';
        ?>
        mode: '000644'
        owner: apache
        group: apache
/etc/cfn/cfn-hup.conf:
    content: !Sub |
        [main]
        stack=${AWS::StackId}
        region=${AWS::Region}
    mode: '000400'
    owner: root
    group: root
/etc/cfn/hooks.d/cfn-auto-reloader.conf:
    content: !Sub |
        [cfn-auto-reloader-hook]
        triggers=post.update
        path=Resources.WebServerHost.Metadata.AWS::CloudFormation::Init
        action=/opt/aws/bin/cfn-init -s ${AWS::StackId} -r WebServerHost --
region ${AWS::Region}
        runas=root
services:
    systemd:
        httpd:
            enabled: 'true'
            ensureRunning: 'true'
        php-fpm:
            enabled: 'true'
            ensureRunning: 'true'
CreationPolicy:
    ResourceSignal:
        Timeout: PT5M
Properties:
    ImageId: '{{resolve:ssm:/aws/service/ami-amazon-linux-latest/al2023-ami-
kernel-6.1-x86_64}}'
    KeyName: !Ref KeyName
    InstanceType: t3.micro
    SecurityGroupIds:

```

```

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

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

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

        # Signal success
        /opt/aws/bin/cfn-signal -e $? --stack ${AWS::StackName} --resource
WebServerHost --region ${AWS::Region}
        echo "Done"
WebServerLogGroup:
    Type: AWS::Logs::LogGroup
    DeletionPolicy: Retain
    UpdateReplacePolicy: Retain
    Properties:
        RetentionInDays: 7
404MetricFilter:
    Type: AWS::Logs::MetricFilter
    Properties:
        LogGroupName: !Ref WebServerLogGroup
        FilterPattern: >-
            [ip, identity, user_id, timestamp, request, status_code = 404, size, ...]
        MetricTransformations:
            - MetricValue: '1'
              MetricNamespace: test/404s
              MetricName: test404Count
BytesTransferredMetricFilter:
    Type: AWS::Logs::MetricFilter
    Properties:
        LogGroupName: !Ref WebServerLogGroup
        FilterPattern: '[ip, identity, user_id, timestamp, request, status_code,
size, ...]'
        MetricTransformations:

```

```
- MetricValue: $size
  MetricNamespace: test/BytesTransferred
  MetricName: testBytesTransferred
```

404Alarm:

```
Type: AWS::CloudWatch::Alarm
```

Properties:

```
AlarmDescription: The number of 404s is greater than 2 over 2 minutes
MetricName: test404Count
Namespace: test/404s
Statistic: Sum
Period: '60'
EvaluationPeriods: '2'
Threshold: '2'
AlarmActions:
  - !Ref AlarmNotificationTopic
ComparisonOperator: GreaterThanThreshold
```

BandwidthAlarm:

```
Type: AWS::CloudWatch::Alarm
```

Properties:

```
AlarmDescription: The average volume of traffic is greater 3500 KB over 10
minutes
MetricName: testBytesTransferred
Namespace: test/BytesTransferred
Statistic: Average
Period: '300'
EvaluationPeriods: '2'
Threshold: '3500'
AlarmActions:
  - !Ref AlarmNotificationTopic
ComparisonOperator: GreaterThanThreshold
```

AlarmNotificationTopic:

```
Type: AWS::SNS::Topic
```

Properties:

```
Subscription:
  - Endpoint: !Ref OperatorEmail
  Protocol: email
```

Outputs:**InstanceId:**

```
Description: The instance ID of the web server
Value: !Ref WebServerHost
```

WebsiteURL:

```
Value: !Sub 'http://${WebServerHost.PublicDnsName}'
Description: URL for the web server
```

PublicIP:

```

    Description: Public IP address of the web server
    Value: !GetAtt WebServerHost.PublicIp
  CloudWatchLogGroupName:
    Description: The name of the CloudWatch log group
    Value: !Ref WebServerLogGroup

```

Enviar registros a CloudWatch Logs desde una instancia Windows

La siguiente plantilla configura CloudWatch Logs para una instancia de Windows 2012 R2.

El agente CloudWatch Logs en Windows (agente SSM en AMI de Windows 2012R2 y Windows 2016) solo envía registros después de iniciarse, para que no se envíen los registros que se generan antes del inicio. Para solucionar esto, la plantilla ayuda a garantizar que el agente comience antes de que se escriban registros:

- Configurando los parámetros del agente como primer elemento de config en cfn-init configSets.
- Utilizando waitAfterCompletion para insertar una pausa tras el comando que inicia el agente.

JSON

```

{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "Sample template that sets up and configures CloudWatch Logs on
Windows 2012R2 instance.",
  "Parameters": {
    "KeyPair": {
      "Description": "Name of an existing EC2 KeyPair to enable RDP access to the
instances",
      "Type": "AWS::EC2::KeyPair::KeyName",
      "ConstraintDescription": "must be the name of an existing EC2 KeyPair."
    },
    "RDPLocation": {
      "Description": "The IP address range that can be used to RDP to the EC2
instances",
      "Type": "String",
      "MinLength": "9",
      "MaxLength": "18",
      "Default": "0.0.0.0/0",
      "AllowedPattern": "(\\d{1,3})\\. (\\d{1,3})\\. (\\d{1,3})\\. (\\d{1,3})/(\\
\\d{1,2})",

```



```

        "ConstraintDescription": "must be a valid IP CIDR range of the form
x.x.x.x/x."
    },
    "OperatorEmail": {
        "Description": "Email address to notify when CloudWatch alarms are
triggered (404 errors)",
        "Type": "String"
    }
},
"Resources": {
    "WebServerSecurityGroup": {
        "Type": "AWS::EC2::SecurityGroup",
        "Properties": {
            "GroupDescription": "Enable HTTP access via port 80 and RDP access via
port 3389",
            "SecurityGroupIngress": [
                {
                    "IpProtocol": "tcp",
                    "FromPort": "80",
                    "ToPort": "80",
                    "CidrIp": "0.0.0.0/0"
                },
                {
                    "IpProtocol": "tcp",
                    "FromPort": "3389",
                    "ToPort": "3389",
                    "CidrIp": {"Ref": "RDPLocation"}
                }
            ]
        }
    },
    "LogRole": {
        "Type": "AWS::IAM::Role",
        "Properties": {
            "AssumeRolePolicyDocument": {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Effect": "Allow",
                        "Principal": {
                            "Service": [
                                "ec2.amazonaws.com"
                            ]
                        }
                    }
                ]
            }
        }
    },

```

```

        "Action": [
            "sts:AssumeRole"
        ]
    },
    ],
    "ManagedPolicyArns": [
        "arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore"
    ],
    "Path": "/",
    "Policies": [
        {
            "PolicyName": "LogRolePolicy",
            "PolicyDocument": {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Effect": "Allow",
                        "Action": [
                            "logs:Create*",
                            "logs:PutLogEvents",
                            "s3:GetObject"
                        ],
                        "Resource": [
                            "arn:aws:logs:*:*:*",
                            "arn:aws:s3:::*"
                        ]
                    }
                ]
            }
        }
    ],
    },
    "LogRoleInstanceProfile": {
        "Type": "AWS::IAM::InstanceProfile",
        "Properties": {
            "Path": "/",
            "Roles": [{"Ref": "LogRole"}]
        }
    },
    "WebServerHost": {
        "Type": "AWS::EC2::Instance",
        "CreationPolicy": {

```

```

        "ResourceSignal": {
            "Timeout": "PT15M"
        }
    },
    "Metadata": {
        "AWS::CloudFormation::Init": {
            "configSets": {
                "config": [
                    "00-ConfigureCWLogs",
                    "01-InstallWebServer",
                    "02-ConfigureApplication",
                    "03-Finalize"
                ]
            },
            "00-ConfigureCWLogs": {
                "files": {
                    "C:\\Program Files\\Amazon\\SSM\\Plugins\\awsCloudWatch\\
\\AWS.EC2.Windows.CloudWatch.json": {
                        "content": {
                            "EngineConfiguration": {
                                "Components": [
                                    {
                                        "FullName":
"\"AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch\"",
                                        "Id": "ApplicationEventLog",
                                        "Parameters": {
                                            "Levels": "7",
                                            "LogName": "Application"
                                        }
                                    },
                                    {
                                        "FullName":
"\"AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch\"",
                                        "Id": "SystemEventLog",
                                        "Parameters": {
                                            "Levels": "7",
                                            "LogName": "System"
                                        }
                                    },
                                    {
                                        "FullName":
"\"AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch\"",
                                        "Id": "SecurityEventLog",
                                        "Parameters": {

```

```

        "Levels": "7",
        "LogName": "Security"
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch",
        "Id": "EC2ConfigLog",
        "Parameters": {
            "CultureName": "en-US",
            "Encoding": "ASCII",
            "Filter": "EC2ConfigLog.txt",
            "LogDirectoryPath": "C:\\Program
Files\\Amazon\\Ec2ConfigService\\Logs",
            "TimeZoneKind": "UTC",
            "TimestampFormat": "yyyy-MM-
ddTHH:mm:ss.fffZ:"
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch",
        "Id": "CfnInitLog",
        "Parameters": {
            "CultureName": "en-US",
            "Encoding": "ASCII",
            "Filter": "cfn-init.log",
            "LogDirectoryPath": "C:\\cfn\\log",
            "TimeZoneKind": "Local",
            "TimestampFormat": "yyyy-MM-dd
HH:mm:ss,fff"
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch",
        "Id": "IISLogs",
        "Parameters": {
            "CultureName": "en-US",
            "Encoding": "UTF-8",
            "Filter": "",
            "LineCount": "3",
            "LogDirectoryPath": "C:\\inetpub\\
logs\\LogFiles\\W3SVC1",

```

```

        "TimeZoneKind": "UTC",
        "TimestampFormat": "yyyy-MM-dd

HH:mm:ss"
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.PerformanceCounterComponent.PerformanceCounterInputComponent,AWS.E
        "Id": "MemoryPerformanceCounter",
        "Parameters": {
            "CategoryName": "Memory",
            "CounterName": "Available MBytes",
            "DimensionName": "",
            "DimensionValue": "",
            "InstanceName": "",
            "MetricName": "Memory",
            "Unit": "Megabytes"
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
        "Id": "CloudWatchApplicationEventLog",
        "Parameters": {
            "AccessKey": "",
            "LogGroup": {"Ref": "LogGroup"},
            "LogStream": "{instance_id}/

ApplicationEventLog",

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

SystemEventLog",

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

```

```

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

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

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

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

```

```

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

IISLogs",

            "Region": {"Ref": "AWS::Region"},
            "SecretKey": ""
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CloudWatch.CloudWatchOutputComponent,AWS.EC2.Windows.CloudWatch",
        "Id": "CloudWatch",
        "Parameters": {
            "AccessKey": "",
            "NameSpace": "Windows/Default",
            "Region": {"Ref": "AWS::Region"},
            "SecretKey": ""
        }
    }
],
"Flows": {
    "Flows": [

"ApplicationEventLog,CloudWatchApplicationEventLog",

"SystemEventLog,CloudWatchSystemEventLog",

"SecurityEventLog,CloudWatchSecurityEventLog",

"EC2ConfigLog,CloudWatchEC2ConfigLog",
"CfnInitLog,CloudWatchCfnInitLog",
"IISLogs,CloudWatchIISLogs",
"MemoryPerformanceCounter,CloudWatch"

    ]
},
    "PollInterval": "00:00:05"
},
    "IsEnabled": true
}
    },
    "commands": {
        "0-enableSSM": {

```

```

        "command": "powershell.exe -Command \"Set-Service -Name
AmazonSSMAgent -StartupType Automatic\" ",
        "waitAfterCompletion": "0"
    },
    "1-restartSSM": {
        "command": "powershell.exe -Command \"Restart-Service
AmazonSSMAgent \"",
        "waitAfterCompletion": "30"
    }
},
"01-InstallWebServer": {
    "commands": {
        "01_install_webserver": {
            "command": "powershell.exe -Command \"Install-
WindowsFeature Web-Server -IncludeAllSubFeature\"",
            "waitAfterCompletion": "0"
        }
    }
},
"02-ConfigureApplication": {
    "files": {
        "c:\\\\Inetpub\\\\wwwroot\\\\index.htm": {
            "content": "<html> <head> <title>Test Application
Page</title> </head> <body> <h1>Congratulations!! Your IIS server is configured.</h1>
</body> </html>"
        }
    }
},
"03-Finalize": {
    "commands": {
        "00_signal_success": {
            "command": {
                "Fn::Sub": "cfn-signal.exe -e 0 --resource
WebServerHost --stack ${AWS::StackName} --region ${AWS::Region}"
            },
            "waitAfterCompletion": "0"
        }
    }
}
},
"Properties": {
    "KeyName": {

```



```

        "Ref": "KeyPair"
    },
    "ImageId": "{{resolve:ssm:/aws/service/ami-windows-latest/
Windows_Server-2012-R2_RTM-English-64Bit-Base}}",
    "InstanceType": "t2.xlarge",
    "SecurityGroupIds": [{"Ref": "WebServerSecurityGroup"}],
    "IamInstanceProfile": {"Ref": "LogRoleInstanceProfile"},
    "UserData": {
        "Fn::Base64": {
            "Fn::Join": [
                "",
                [
                    "<script>\n",
                    "wmic product where \"description='Amazon SSM Agent' \"
uninstall\n",
                    "wmic product where \"description='aws-cfn-bootstrap'
\" uninstall \n",
                    "start /wait c:\\\\Windows\\system32\\msiexec /passive /
qn /i https://s3.amazonaws.com/cloudformation-examples/aws-cfn-bootstrap-win64-
latest.msi\n",
                    "powershell.exe -Command \"iwr https://
s3.amazonaws.com/ec2-downloads-windows/SSMAgent/latest/windows_amd64/
AmazonSSMAgentSetup.exe -UseBasicParsing -OutFile C:\\\\AmazonSSMAgentSetup.exe\"\\n",
                    "start /wait C:\\\\AmazonSSMAgentSetup.exe /install /
quiet\n",
                    "cfn-init.exe -v -c config -s ", {"Ref":
"AWS::StackName"}, " --resource WebServerHost --region ", {"Ref": "AWS::Region"}, "
\n",
                    "</script>\n"
                ]
            ]
        }
    },
    "LogGroup": {
        "Type": "AWS::Logs::LogGroup",
        "Properties": {
            "RetentionInDays": 7
        }
    },
    "404MetricFilter": {
        "Type": "AWS::Logs::MetricFilter",
        "Properties": {

```

```

        "LogGroupName": {"Ref": "LogGroup"},
        "FilterPattern": "[timestamps, serverip, method, uri, query, port,
dash, clientip, useragent, status_code = 404, ...]",
        "MetricTransformations": [
            {
                "MetricValue": "1",
                "MetricNamespace": "test/404s",
                "MetricName": "test404Count"
            }
        ]
    },
    "404Alarm": {
        "Type": "AWS::CloudWatch::Alarm",
        "Properties": {
            "AlarmDescription": "The number of 404s is greater than 2 over 2
minutes",
            "MetricName": "test404Count",
            "Namespace": "test/404s",
            "Statistic": "Sum",
            "Period": "60",
            "EvaluationPeriods": "2",
            "Threshold": "2",
            "AlarmActions": [{"Ref": "AlarmNotificationTopic"}],
            "ComparisonOperator": "GreaterThanThreshold"
        }
    },
    "AlarmNotificationTopic": {
        "Type": "AWS::SNS::Topic",
        "Properties": {
            "Subscription": [{"Endpoint": {"Ref": "OperatorEmail"}, "Protocol":
"email"}]
        }
    },
    "Outputs": {
        "InstanceId": {
            "Description": "The instance ID of the web server",
            "Value": {"Ref": "WebServerHost"}
        },
        "WebsiteURL": {
            "Value": {"Fn::Sub": "http://${WebServerHost.PublicDnsName}"},
            "Description": "URL for the web server"
        }
    },

```

```

    "PublicIP": {
      "Description": "Public IP address of the web server",
      "Value": {"Fn::GetAtt": ["WebServerHost", "PublicIp"]}
    },
    "CloudWatchLogGroupName": {
      "Description": "The name of the CloudWatch log group",
      "Value": {"Ref": "LogGroup"}
    }
  }
}

```

YAML

```

AWSTemplateFormatVersion: 2010-09-09
Description: >-
  Sample template that sets up and configures CloudWatch Logs on Windows 2012R2
  instance.
Parameters:
  KeyPair:
    Description: Name of an existing EC2 KeyPair to enable RDP access to the instances
    Type: AWS::EC2::KeyPair::KeyName
    ConstraintDescription: must be the name of an existing EC2 KeyPair.
  RDPLocation:
    Description: The IP address range that can be used to RDP to the EC2 instances
    Type: String
    MinLength: '9'
    MaxLength: '18'
    Default: 0.0.0.0/0
    AllowedPattern: '(\d{1,3})\.\d{1,3})\.\d{1,3})\.\d{1,3})/(\d{1,2})'
    ConstraintDescription: must be a valid IP CIDR range of the form x.x.x.x/x.
  OperatorEmail:
    Description: Email address to notify when CloudWatch alarms are triggered (404
    errors)
    Type: String
Resources:
  WebServerSecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupDescription: Enable HTTP access via port 80 and RDP access via port 3389
      SecurityGroupIngress:
        - IpProtocol: tcp
          FromPort: '80'
          ToPort: '80'

```

```
    CidrIp: 0.0.0.0/0
  - IpProtocol: tcp
    FromPort: '3389'
    ToPort: '3389'
    CidrIp: !Ref RDPLocation
LogRole:
  Type: AWS::IAM::Role
  Properties:
    AssumeRolePolicyDocument:
      Version: 2012-10-17
      Statement:
        - Effect: Allow
          Principal:
            Service:
              - ec2.amazonaws.com
          Action:
            - 'sts:AssumeRole'
    ManagedPolicyArns:
      - 'arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore'
  Path: /
  Policies:
    - PolicyName: LogRolePolicy
      PolicyDocument:
        Version: 2012-10-17
        Statement:
          - Effect: Allow
            Action:
              - 'logs:Create*'
              - 'logs:PutLogEvents'
              - 's3:GetObject'
            Resource:
              - 'arn:aws:logs:*:*:*'
              - 'arn:aws:s3::*:*'
LogRoleInstanceProfile:
  Type: AWS::IAM::InstanceProfile
  Properties:
    Path: /
    Roles:
      - !Ref LogRole
WebServerHost:
  Type: AWS::EC2::Instance
  CreationPolicy:
    ResourceSignal:
      Timeout: PT15M
```

```

Metadata:
  'AWS::CloudFormation::Init':
    configSets:
      config:
        - 00-ConfigureCWLogs
        - 01-InstallWebServer
        - 02-ConfigureApplication
        - 03-Finalize
    00-ConfigureCWLogs:
      files:
        'C:\Program Files\Amazon\SSM\Plugins\awsCloudWatch
\AWS.EC2.Windows.CloudWatch.json':
          content: !Sub |
            {
              "EngineConfiguration": {
                "Components": [
                  {
                    "FullName":
"AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
                    "Id": "ApplicationEventLog",
                    "Parameters": {
                      "Levels": "7",
                      "LogName": "Application"
                    }
                  },
                  {
                    "FullName":
"AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
                    "Id": "SystemEventLog",
                    "Parameters": {
                      "Levels": "7",
                      "LogName": "System"
                    }
                  },
                  {
                    "FullName":
"AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
                    "Id": "SecurityEventLog",
                    "Parameters": {
                      "Levels": "7",
                      "LogName": "Security"
                    }
                  }
                ]
              }
            }

```

```

        "FullName":
"AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch",
        "Id": "EC2ConfigLog",
        "Parameters": {
            "CultureName": "en-US",
            "Encoding": "ASCII",
            "Filter": "EC2ConfigLog.txt",
            "LogDirectoryPath": "C:\\\\Program Files\\Amazon\\
\\Ec2ConfigService\\Logs",
            "TimeZoneKind": "UTC",
            "TimestampFormat": "yyyy-MM-ddTHH:mm:ss.fffZ:"
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch",
        "Id": "CfnInitLog",
        "Parameters": {
            "CultureName": "en-US",
            "Encoding": "ASCII",
            "Filter": "cfn-init.log",
            "LogDirectoryPath": "C:\\\\cfn\\log",
            "TimeZoneKind": "Local",
            "TimestampFormat": "yyyy-MM-dd HH:mm:ss,fff"
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch",
        "Id": "IISLogs",
        "Parameters": {
            "CultureName": "en-US",
            "Encoding": "UTF-8",
            "Filter": "",
            "LineCount": "3",
            "LogDirectoryPath": "C:\\\\inetpub\\logs\\LogFiles\\
\\W3SVC1",
            "TimeZoneKind": "UTC",
            "TimestampFormat": "yyyy-MM-dd HH:mm:ss"
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.PerformanceCounterComponent.PerformanceCounterInputComponent,AWS.E

```

```

        "Id": "MemoryPerformanceCounter",
        "Parameters": {
            "CategoryName": "Memory",
            "CounterName": "Available MBytes",
            "DimensionName": "",
            "DimensionValue": "",
            "InstanceName": "",
            "MetricName": "Memory",
            "Unit": "Megabytes"
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
        "Id": "CloudWatchApplicationEventLog",
        "Parameters": {
            "AccessKey": "",
            "LogGroup": "${LogGroup}",
            "LogStream": "{instance_id}/ApplicationEventLog",
            "Region": "${AWS::Region}",
            "SecretKey": ""
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
        "Id": "CloudWatchSystemEventLog",
        "Parameters": {
            "AccessKey": "",
            "LogGroup": "${LogGroup}",
            "LogStream": "{instance_id}/SystemEventLog",
            "Region": "${AWS::Region}",
            "SecretKey": ""
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
        "Id": "CloudWatchSecurityEventLog",
        "Parameters": {
            "AccessKey": "",
            "LogGroup": "${LogGroup}",
            "LogStream": "{instance_id}/SecurityEventLog",
            "Region": "${AWS::Region}",

```

```

        "SecretKey": ""
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
        "Id": "CloudWatchEC2ConfigLog",
        "Parameters": {
            "AccessKey": "",
            "LogGroup": "${LogGroup}",
            "LogStream": "{instance_id}/EC2ConfigLog",
            "Region": "${AWS::Region}",
            "SecretKey": ""
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
        "Id": "CloudWatchCfnInitLog",
        "Parameters": {
            "AccessKey": "",
            "LogGroup": "${LogGroup}",
            "LogStream": "{instance_id}/CfnInitLog",
            "Region": "${AWS::Region}",
            "SecretKey": ""
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
        "Id": "CloudWatchIISLogs",
        "Parameters": {
            "AccessKey": "",
            "LogGroup": "${LogGroup}",
            "LogStream": "{instance_id}/IISLogs",
            "Region": "${AWS::Region}",
            "SecretKey": ""
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CloudWatch.CloudWatchOutputComponent,AWS.EC2.Windows.CloudWatch",
        "Id": "CloudWatch",
        "Parameters": {

```



```

        "AccessKey": "",
        "NameSpace": "Windows/Default",
        "Region": "${AWS::Region}",
        "SecretKey": ""
    }
}
],
"Flows": {
    "Flows": [
        "ApplicationEventLog,CloudWatchApplicationEventLog",
        "SystemEventLog,CloudWatchSystemEventLog",
        "SecurityEventLog,CloudWatchSecurityEventLog",
        "EC2ConfigLog,CloudWatchEC2ConfigLog",
        "CfnInitLog,CloudWatchCfnInitLog",
        "IISLogs,CloudWatchIISLogs",
        "MemoryPerformanceCounter,CloudWatch"
    ]
},
"PollInterval": "00:00:05"
},
"IsEnabled": true
}
commands:
  0-enableSSM:
    command: >-
      powershell.exe -Command "Set-Service -Name AmazonSSMAgent
        -StartupType Automatic"
    waitAfterCompletion: '0'
  1-restartSSM:
    command: powershell.exe -Command "Restart-Service AmazonSSMAgent "
    waitAfterCompletion: '30'
01-InstallWebServer:
  commands:
    01_install_webserver:
      command: >-
        powershell.exe -Command "Install-WindowsFeature Web-Server
          -IncludeAllSubFeature"
      waitAfterCompletion: '0'
02-ConfigureApplication:
  files:
    'c:\Inetpub\wwwroot\index.htm':
      content: >-
        <html> <head> <title>Test Application Page</title> </head>
        <body> <h1>Congratulations !! Your IIS server is

```

```

        configured.</h1> </body> </html>
    03-Finalize:
        commands:
            00_signal_success:
                command: !Sub >-
                    cfn-signal.exe -e 0 --resource WebServerHost --stack
                        ${AWS::StackName} --region ${AWS::Region}
                waitAfterCompletion: '0'
    Properties:
        KeyName: !Ref KeyPair
        ImageId: "{{resolve:ssm:/aws/service/ami-windows-latest/Windows_Server-2012-
R2_RTM-English-64Bit-Base}}"
        InstanceType: t2.xlarge
        SecurityGroupIds:
            - !Ref WebServerSecurityGroup
        IamInstanceProfile: !Ref LogRoleInstanceProfile
        UserData: !Base64
            'Fn::Sub': >
                <script>

                    wmic product where "description='Amazon SSM Agent' " uninstall

                    wmic product where "description='aws-cfn-bootstrap' " uninstall

                    start /wait c:\\Windows\\system32\\msiexec /passive /qn /i
                    https://s3.amazonaws.com/cloudformation-examples/aws-cfn-bootstrap-win64-
latest.msi

                    powershell.exe -Command "iwr
                    https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/latest/windows_amd64/
AmazonSSMAgentSetup.exe
                    -UseBasicParsing -OutFile C:\\AmazonSSMAgentSetup.exe"

                    start /wait C:\\AmazonSSMAgentSetup.exe /install /quiet

                    cfn-init.exe -v -c config -s ${AWS::StackName} --resource
                    WebServerHost --region ${AWS::Region}

                </script>
        LogGroup:
            Type: AWS::Logs::LogGroup
            Properties:
                RetentionInDays: 7
        404MetricFilter:

```

```
Type: AWS::Logs::MetricFilter
```

```
Properties:
```

```
  LogGroupName: !Ref LogGroup
```

```
  FilterPattern: >-
```

```
    [timestamps, serverip, method, uri, query, port, dash, clientip,  
    useragent, status_code = 404, ...]
```

```
  MetricTransformations:
```

```
    - MetricValue: '1'
```

```
      MetricNamespace: test/404s
```

```
      MetricName: test404Count
```

```
404Alarm:
```

```
Type: AWS::CloudWatch::Alarm
```

```
Properties:
```

```
  AlarmDescription: The number of 404s is greater than 2 over 2 minutes
```

```
  MetricName: test404Count
```

```
  Namespace: test/404s
```

```
  Statistic: Sum
```

```
  Period: '60'
```

```
  EvaluationPeriods: '2'
```

```
  Threshold: '2'
```

```
  AlarmActions:
```

```
    - !Ref AlarmNotificationTopic
```

```
  ComparisonOperator: GreaterThanThreshold
```

```
AlarmNotificationTopic:
```

```
Type: AWS::SNS::Topic
```

```
Properties:
```

```
  Subscription:
```

```
    - Endpoint: !Ref OperatorEmail
```

```
      Protocol: email
```

```
Outputs:
```

```
  InstanceId:
```

```
    Description: The instance ID of the web server
```

```
    Value: !Ref WebServerHost
```

```
  WebsiteURL:
```

```
    Value: !Sub 'http://${WebServerHost.PublicDnsName}'
```

```
    Description: URL for the web server
```

```
  PublicIP:
```

```
    Description: Public IP address of the web server
```

```
    Value: !GetAtt
```

```
      - WebServerHost
```

```
      - PublicIp
```

```
CloudWatchLogGroupName:
```

```
  Description: The name of the CloudWatch log group
```

Value: !Ref LogGroup

Véase también

Para obtener más información sobre los recursos de CloudWatch Logs, consulte [AWS::Logs::LogGroup](#) o [AWS::Logs::MetricFilter](#).

Fragmentos de plantillas de Amazon DynamoDB

Temas

- [Application Auto Scaling con una tabla de Amazon DynamoDB](#)
- [Véase también](#)

Application Auto Scaling con una tabla de Amazon DynamoDB

En este ejemplo se configura Application Auto Scaling para un recurso de `AWS::DynamoDB::Table`. La plantilla define una política de escalado de `TargetTrackingScaling` que escala el desempeño de `WriteCapacityUnits` para la tabla.

JSON

```
{
  "Resources": {
    "DDBTable": {
      "Type": "AWS::DynamoDB::Table",
      "Properties": {
        "AttributeDefinitions": [
          {
            "AttributeName": "ArtistId",
            "AttributeType": "S"
          },
          {
            "AttributeName": "Concert",
            "AttributeType": "S"
          },
          {
            "AttributeName": "TicketSales",
            "AttributeType": "S"
          }
        ],
        "KeySchema": [
```

```

        {
            "AttributeName": "ArtistId",
            "KeyType": "HASH"
        },
        {
            "AttributeName": "Concert",
            "KeyType": "RANGE"
        }
    ],
    "GlobalSecondaryIndexes": [
        {
            "IndexName": "GSI",
            "KeySchema": [
                {
                    "AttributeName": "TicketSales",
                    "KeyType": "HASH"
                }
            ],
            "Projection": {
                "ProjectionType": "KEYS_ONLY"
            },
            "ProvisionedThroughput": {
                "ReadCapacityUnits": 5,
                "WriteCapacityUnits": 5
            }
        }
    ],
    "ProvisionedThroughput": {
        "ReadCapacityUnits": 5,
        "WriteCapacityUnits": 5
    }
},
"WriteCapacityScalableTarget": {
    "Type": "AWS::ApplicationAutoScaling::ScalableTarget",
    "Properties": {
        "MaxCapacity": 15,
        "MinCapacity": 5,
        "ResourceId": {
            "Fn::Join": [
                "/",
                [
                    "table",
                    {

```

```

        "Ref": "DDBTable"
      }
    ]
  ],
  "RoleARN" : { "Fn::Sub" : "arn:aws:iam::${AWS::AccountId}:role/aws-service-role/dynamodb.application-autoscaling.amazonaws.com/AWSServiceRoleForApplicationAutoScaling_DynamoDBTable" },
  "ScalableDimension": "dynamodb:table:WriteCapacityUnits",
  "ServiceNamespace": "dynamodb"
}
},
"WriteScalingPolicy": {
  "Type": "AWS::ApplicationAutoScaling::ScalingPolicy",
  "Properties": {
    "PolicyName": "WriteAutoScalingPolicy",
    "PolicyType": "TargetTrackingScaling",
    "ScalingTargetId": {
      "Ref": "WriteCapacityScalableTarget"
    },
    "TargetTrackingScalingPolicyConfiguration": {
      "TargetValue": 50,
      "ScaleInCooldown": 60,
      "ScaleOutCooldown": 60,
      "PredefinedMetricSpecification": {
        "PredefinedMetricType": "DynamoDBWriteCapacityUtilization"
      }
    }
  }
}
}
}
}

```

YAML

```
Resources:
  DDBTable:
    Type: AWS::DynamoDB::Table
    Properties:
      AttributeDefinitions:
        - AttributeName: "ArtistId"
          AttributeType: "S"
        - AttributeName: "Concert"
```

```

    AttributeType: "S"
  - AttributeName: "TicketSales"
    AttributeType: "S"
  KeySchema:
  - AttributeName: "ArtistId"
    KeyType: "HASH"
  - AttributeName: "Concert"
    KeyType: "RANGE"
  GlobalSecondaryIndexes:
  - IndexName: "GSI"
    KeySchema:
      - AttributeName: "TicketSales"
        KeyType: "HASH"
    Projection:
      ProjectionType: "KEYS_ONLY"
    ProvisionedThroughput:
      ReadCapacityUnits: 5
      WriteCapacityUnits: 5
  ProvisionedThroughput:
    ReadCapacityUnits: 5
    WriteCapacityUnits: 5
  WriteCapacityScalableTarget:
    Type: AWS::ApplicationAutoScaling::ScalableTarget
    Properties:
      MaxCapacity: 15
      MinCapacity: 5
      ResourceId: !Join
        - /
        - - table
          - !Ref DDBTable
      RoleARN:
        Fn::Sub: 'arn:aws:iam::${AWS::AccountId}:role/aws-
service-role/dynamodb.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_DynamoDBTable'
      ScalableDimension: dynamodb:table:WriteCapacityUnits
      ServiceNamespace: dynamodb
  WriteScalingPolicy:
    Type: AWS::ApplicationAutoScaling::ScalingPolicy
    Properties:
      PolicyName: WriteAutoScalingPolicy
      PolicyType: TargetTrackingScaling
      ScalingTargetId: !Ref WriteCapacityScalableTarget
      TargetTrackingScalingPolicyConfiguration:
        TargetValue: 50.0

```

```
ScaleInCooldown: 60
ScaleOutCooldown: 60
PredefinedMetricSpecification:
  PredefinedMetricType: DynamoDBWriteCapacityUtilization
```

Véase también

Para obtener más información, consulte [How to use CloudFormation to configure auto scaling for DynamoDB tables and indexes](#) en el blog de bases de datos de AWS.

Para obtener más información sobre recursos de DynamoDB, consulte [AWS::DynamoDB::Table](#).

Fragmentos de plantillas de Amazon EC2 CloudFormation

Amazon EC2 proporciona capacidad de computación escalable en la Nube de AWS. Puede usar Amazon EC2 para lanzar tantos servidores virtuales como necesite, configurar la seguridad y las redes, y administrar el almacenamiento. Estos servidores virtuales, conocidos como instancias, pueden ejecutar una variedad de sistemas operativos y aplicaciones, y se pueden personalizar para cumplir con sus requisitos específicos. Amazon EC2 permite escalar o desescalar verticalmente para afrontar los cambios que se produzcan en los requisitos.

Puede definir y aprovisionar instancias de Amazon EC2 como parte de su infraestructura mediante plantillas de CloudFormation. Las plantillas facilitan la administración y la automatización de la implementación de los recursos de Amazon EC2 de forma repetible y coherente.

Los siguientes fragmentos de plantilla de ejemplo describen los recursos o componentes de CloudFormation para Amazon EC2. Estos fragmentos están diseñados para integrarse en una plantilla y no están pensados para ejecutarse de forma independiente.

Categorías de fragmentos

- [Configuración de instancias de Amazon EC2 con CloudFormation](#)
- [Creación de plantillas de inicialización con CloudFormation](#)
- [Administración de grupos de seguridad con CloudFormation](#)
- [Asignación y asociación de una dirección IP elástica con CloudFormation](#)
- [Configuración de los recursos de Amazon VPC con CloudFormation](#)

Configuración de instancias de Amazon EC2 con CloudFormation

Los siguientes fragmentos muestran cómo configurar las instancias de Amazon EC2 mediante CloudFormation.

Categorías de fragmentos

- [Configuración general de Amazon EC2](#)
- [Especificación de las asignaciones de dispositivos de bloques para la instancia.](#)

Configuración general de Amazon EC2

Los siguientes fragmentos muestran las configuraciones generales de las instancias de Amazon EC2 mediante CloudFormation.

Fragmentos de código de ejemplo

- [Creación de una instancia de Amazon EC2 en una zona de disponibilidad específica](#)
- [Configuración de una instancia de Amazon EC2 con un volumen de EBS y datos de usuario](#)
- [Definición del nombre de la tabla de DynamoDB en los datos de usuario para el lanzamiento de la instancia de Amazon EC2](#)
- [Creación de un volumen de Amazon EBS con DeletionPolicy](#)

Creación de una instancia de Amazon EC2 en una zona de disponibilidad específica

En el siguiente fragmento se crea una instancia de Amazon EC2 en la zona de disponibilidad específica con un recurso [AWS::EC2::Instance](#). El código de la zona de disponibilidad es el código de la región seguido de un identificador de letra. Puede lanzar una instancia en una sola zona de disponibilidad.

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
```

Configuración de una instancia de Amazon EC2 con un volumen de EBS y datos de usuario

El siguiente fragmento crea una instancia de Amazon EC2 con una etiqueta, un volumen de EBS y datos de usuario. Utiliza un recurso [AWS::EC2::Instance](#). En la misma plantilla, debe definir un recurso [AWS::EC2::SecurityGroup](#), un recurso [AWS::SNS::Topic](#) y un recurso [AWS::EC2::Volume](#). Se debe definir KeyName en la sección Parameters de la plantilla.

Las etiquetas permiten clasificar los recursos de AWS de diversas maneras, por ejemplo, según su finalidad, propietario o entorno. Los datos de usuario permiten aprovisionar scripts o datos personalizados a una instancia durante el lanzamiento. Estos datos facilitan la automatización de tareas, la configuración del software, la instalación de paquetes y otras acciones en una instancia durante la inicialización.

Para obtener más información sobre el etiquetado de recursos, consulte [Etiquetar los recursos de Amazon EC2](#) en la Guía del usuario de Amazon EC2.

Para obtener información sobre los datos de usuario, consulte [Uso de los metadatos de la instancia para administrar su instancia de EC2](#) en la Guía del usuario de Amazon EC2.

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

```

Definición del nombre de la tabla de DynamoDB en los datos de usuario para el lanzamiento de la instancia de Amazon EC2

El siguiente fragmento crea una instancia de Amazon EC2 y define un nombre de tabla de DynamoDB en los datos de usuario para pasarlo a la instancia en el momento del lanzamiento. Utiliza un recurso [AWS::EC2::Instance](#). Puede definir parámetros o valores dinámicos en los datos del usuario para transferirlos a una instancia de EC2 en el momento del lanzamiento.

Para obtener más información sobre los datos de usuario, consulte [Uso de los metadatos de la instancia para administrar su instancia de EC2](#) en la Guía del usuario de Amazon EC2.

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
```

Creación de un volumen de Amazon EBS con **DeletionPolicy**

Los siguientes fragmentos crean un volumen de Amazon EBS mediante un recurso [AWS::EC2::Volume](#) de Amazon EC2. Puede usar las propiedades `Size` o `SnapshotID` para definir el volumen, pero no ambas. Un atributo `DeletionPolicy` está configurado para crear una instantánea del volumen cuando se elimina la pila.

Para obtener más información acerca del atributo `DeletionPolicy`, consulte [Atributo DeletionPolicy](#).

Para más información sobre la creación de una instantánea en EBS, consulte [Creación de un volumen de Amazon EBS](#).

JSON

Este fragmento de código crea un volumen de Amazon EBS con un tamaño especificado. El tamaño está establecido en 10, pero puede ajustarlo según sea necesario. El recurso [AWS::EC2::Volume](#) le permite especificar el tamaño o el ID de una instantánea, pero no ambos.

```
"MyEBSVolume": {  
  "Type": "AWS::EC2::Volume",  
  "Properties": {  
    "Size": "10",  
    "AvailabilityZone": {  
      "Ref": "AvailabilityZone"  
    }  
  },  
  "DeletionPolicy": "Snapshot"  
}
```

Este fragmento de código crea un volumen de Amazon EBS con un ID de instantánea proporcionado. El recurso [AWS::EC2::Volume](#) le permite especificar el tamaño o el ID de una instantánea, pero no ambos.

```
"MyEBSVolume": {
```

```
"Type": "AWS::EC2::Volume",
"Properties": {
  "SnapshotId" : "snap-1234567890abcdef0",
  "AvailabilityZone": {
    "Ref": "AvailabilityZone"
  }
},
"DeletionPolicy": "Snapshot"
}
```

YAML

Este fragmento de código crea un volumen de Amazon EBS con un tamaño especificado. El tamaño está establecido en 10, pero puede ajustarlo según sea necesario. El recurso [AWS::EC2::Volume](#) le permite especificar el tamaño o el ID de una instantánea, pero no ambos.

```
MyEBSVolume:
  Type: AWS::EC2::Volume
  Properties:
    Size: 10
    AvailabilityZone:
      Ref: AvailabilityZone
  DeletionPolicy: Snapshot
```

Este fragmento de código crea un volumen de Amazon EBS con un ID de instantánea proporcionado. El recurso [AWS::EC2::Volume](#) le permite especificar el tamaño o el ID de una instantánea, pero no ambos.

```
MyEBSVolume:
  Type: AWS::EC2::Volume
  Properties:
    SnapshotId: snap-1234567890abcdef0
    AvailabilityZone:
      Ref: AvailabilityZone
  DeletionPolicy: Snapshot
```

Especificación de las asignaciones de dispositivos de bloques para la instancia.

Una asignación de dispositivos de bloques define los dispositivos de bloques, que incluye volúmenes de almacén de instancias y volúmenes de EBS para asociar a la instancia. Puede especificar una asignación de dispositivos de bloques al crear una AMI para que todas las instancias que se lancen

desde la AMI utilicen dicha asignación. También puede especificar una asignación de dispositivos de bloques cuando se lanza una instancia, de forma que esta asignación anula la especificada en la AMI desde la que se lanzó la instancia.

Puede usar los siguientes fragmentos de plantilla para especificar las asignaciones de dispositivos de bloques para sus volúmenes de EBS o de almacén de instancias mediante la propiedad `BlockDeviceMappings` de un recurso de [AWS::EC2::Instance](#).

Para obtener más información acerca de la asignación de dispositivos de bloques, consulte [Asignaciones de dispositivos de bloques para volúmenes en instancias de Amazon EC2](#) en la Guía del usuario de Amazon EC2.

Escenarios

- [Especificación de las asignaciones de dispositivos de bloques para dos volúmenes de EBS](#)
- [Cualquier entrada de asignación de dispositivos de bloques para el almacén de instancias.](#)

Especificación de las asignaciones de dispositivos de bloques para dos volúmenes de EBS

JSON

```
"Ec2Instance": {
  "Type": "AWS::EC2::Instance",
  "Properties": {
    "ImageId": "{{resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2}}",
    "KeyName": { "Ref": "KeyName" },
    "InstanceType": { "Ref": "InstanceType" },
    "SecurityGroups": [{ "Ref": "Ec2SecurityGroup" }],
    "BlockDeviceMappings": [
      {
        "DeviceName": "/dev/sda1",
        "Ebs": { "VolumeSize": "50" }
      },
      {
        "DeviceName": "/dev/sdm",
        "Ebs": { "VolumeSize": "100" }
      }
    ]
  }
}
```

YAML

```

EC2Instance:
  Type: AWS::EC2::Instance
  Properties:
    ImageId: '{{resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-
x86_64-gp2}}'
    KeyName: !Ref KeyName
    InstanceType: !Ref InstanceType
    SecurityGroups:
      - !Ref Ec2SecurityGroup
    BlockDeviceMappings:
      -
        DeviceName: /dev/sda1
        Ebs:
          VolumeSize: 50
      -
        DeviceName: /dev/sdm
        Ebs:
          VolumeSize: 100

```

Cualquier entrada de asignación de dispositivos de bloques para el almacén de instancias.

JSON

```

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

```


YAML

```
EC2Instance:
  Type: AWS::EC2::Instance
  Properties:
    ImageId: '{{resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-
gp2}}'
    KeyName: !Ref KeyName
    InstanceType: !Ref InstanceType
    SecurityGroups:
      - !Ref Ec2SecurityGroup
    BlockDeviceMappings:
      - DeviceName: /dev/sdc
        VirtualName: ephemeral0
```

Creación de plantillas de inicialización con CloudFormation

En esta sección se proporciona un ejemplo de creación de una plantilla de inicialización de Amazon EC2 mediante CloudFormation. Las plantillas de lanzamiento le permiten crear plantillas para configurar y aprovisionar instancias de Amazon EC2 en AWS. Puede utilizar plantillas de lanzamiento para almacenar parámetros de lanzamiento con el objetivo de no tener que especificarlos cada vez que lance una instancia. Para obtener más ejemplos, consulte la sección [Ejemplos](#) del recurso de `AWS::EC2::LaunchTemplate`.

Para obtener más información sobre las plantillas de inicialización, consulte [Almacenamiento de parámetros de inicialización de instancias en plantillas de inicialización de Amazon EC2](#) en la Guía del usuario de Amazon EC2.

Para obtener información acerca de cómo crear una plantilla de inicialización para utilizarla con los grupos de escalado automático, consulte [Plantillas de inicialización de escalado automático](#) en la Guía del usuario de Amazon EC2 Auto Scaling.

Categorías de fragmentos

- [Creación de una plantilla de lanzamiento que especifique grupos de seguridad, etiquetas, datos del usuario y un rol de IAM](#)

Creación de una plantilla de lanzamiento que especifique grupos de seguridad, etiquetas, datos del usuario y un rol de IAM

En este fragmento se muestra un recurso [AWS::EC2::LaunchTemplate](#) que contiene la información de configuración para lanzar una instancia. Los usuarios especifican valores para las propiedades ImageId, InstanceType, SecurityGroups, UserData y TagSpecifications. La propiedad SecurityGroups especifica un grupo de seguridad de EC2 existente y un nuevo grupo de seguridad. La función Ref obtiene el ID del recurso myNewEC2SecurityGroup de [AWS::EC2::SecurityGroup](#) que se declara en otra parte de la plantilla de pila.

La plantilla de lanzamiento incluye una sección para datos de usuario personalizados. En esta sección, puede pasar tareas de configuración y scripts que se ejecutan cuando se lanza una instancia. En este ejemplo, los datos de usuario instalan el agente de AWS Systems Manager e inicia el agente.

La plantilla de lanzamiento también incluye un rol de IAM que permite a las aplicaciones que se ejecutan en instancias llevar a cabo acciones en su nombre. En este ejemplo, se muestra un recurso [AWS::IAM::Role](#) para la plantilla de lanzamiento, que utiliza la propiedad IamInstanceProfile para especificar el rol de IAM. La función Ref obtiene el nombre del recurso myInstanceProfile de [AWS::IAM::InstanceProfile](#). Para configurar los permisos del rol de IAM, debe especificar un valor para la propiedad 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
```

Administración de grupos de seguridad con CloudFormation

Los siguientes fragmentos muestran cómo usar CloudFormation para administrar los grupos de seguridad y las instancias de Amazon EC2 para controlar el acceso a sus recursos de AWS.

Categorías de fragmentos

- [Asociación de una instancia de Amazon EC2 con un grupo de seguridad](#)
- [Creación de grupos de seguridad con reglas de entrada](#)
- [Creación de un equilibrador de carga elástica con una regla de ingreso de grupos de seguridad](#)

Asociación de una instancia de Amazon EC2 con un grupo de seguridad

Los siguientes fragmentos de ejemplo muestran cómo asociar una instancia de Amazon EC2 a un grupo de seguridad de Amazon VPC predeterminado mediante CloudFormation.

Fragmentos de código de ejemplo

- [Asociación de una instancia de Amazon EC2 con un grupo de seguridad de VPN predeterminado](#)
- [Creación de una instancia de Amazon EC2 con un volumen y un grupo de seguridad adjuntos](#)

Asociación de una instancia de Amazon EC2 con un grupo de seguridad de VPN predeterminado

El siguiente fragmento crea una VPC de Amazon, una subred dentro de la VPC y una instancia de Amazon EC2. La VPC se crea con un recurso [AWS::EC2::VPC](#). El rango de direcciones IP de la VPC se define en la plantilla más grande y el parámetro MyVPCCIDRRange hace referencia a él.

Se crea una subred dentro de la VPC con un recurso de subred [AWS::EC2::Subnet](#). La subred está asociada a la VPC, a la que se hace referencia como MyVPC.

Se lanza una instancia de EC2 dentro de la VPC y la subred mediante un recurso [AWS::EC2::Instance](#). Este recurso especifica la imagen de máquina de Amazon (AMI) que se utilizará para lanzar la instancia, la subred en la que se ejecutará la instancia y el grupo de seguridad que se asociará a la instancia. ImageId utiliza un parámetro de Systems Manager para recuperar de forma dinámica la AMI más reciente de Amazon Linux 2.

El ID del grupo de seguridad se obtiene mediante la función `Fn::GetAtt`, que recupera el grupo de seguridad predeterminado del recurso `MyVPC`.

La instancia se coloca dentro del recurso `MySubnet` definido en el fragmento.

Al crear una VPC con CloudFormation, AWS crea automáticamente recursos predeterminados dentro de la VPC, incluido un grupo de seguridad predeterminado. Sin embargo, al definir una VPC dentro de una plantilla de CloudFormation, es posible que no tenga acceso a los ID de estos recursos predeterminados al crear la plantilla. Para acceder a los recursos predeterminados especificados en la plantilla y utilizarlos, puede utilizar funciones intrínsecas, como `Fn::GetAtt`. Esta función le permite trabajar con los recursos predeterminados que CloudFormation crea automáticamente.

JSON

```
"MyVPC": {
  "Type": "AWS::EC2::VPC",
  "Properties": {
    "CidrBlock": {
      "Ref": "MyVPCCIDRRange"
    },
    "EnableDnsSupport": false,
    "EnableDnsHostnames": false,
    "InstanceTenancy": "default"
  }
},
"MySubnet": {
  "Type": "AWS::EC2::Subnet",
  "Properties": {
    "CidrBlock": {
      "Ref": "MyVPCCIDRRange"
    },
    "VpcId": {
      "Ref": "MyVPC"
    }
  }
},
"MyInstance": {
  "Type": "AWS::EC2::Instance",
  "Properties": {
    "ImageId": "{{resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2}}",
    "SecurityGroupIds": [
      {
```

```

        "Fn::GetAtt": [
            "MyVPC",
            "DefaultSecurityGroup"
        ]
    },
    "SubnetId": {
        "Ref": "MySubnet"
    }
}
}

```

YAML

```

MyVPC:
  Type: AWS::EC2::VPC
  Properties:
    CidrBlock:
      Ref: MyVPCCIDRRange
    EnableDnsSupport: false
    EnableDnsHostnames: false
    InstanceTenancy: default
MySubnet:
  Type: AWS::EC2::Subnet
  Properties:
    CidrBlock:
      Ref: MyVPCCIDRRange
    VpcId:
      Ref: MyVPC
MyInstance:
  Type: AWS::EC2::Instance
  Properties:
    ImageId: '{{resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-
gp2}}'
    SecurityGroupIds:
      - Fn::GetAtt:
          - MyVPC
          - DefaultSecurityGroup
    SubnetId:
      Ref: MySubnet

```


Creación de una instancia de Amazon EC2 con un volumen y un grupo de seguridad adjuntos

El siguiente fragmento de código crea una instancia de Amazon EC2 con un recurso [AWS::EC2::Instance](#), que se inicia desde una AMI designada. La instancia está asociada a un grupo de seguridad que permite el tráfico SSH entrante en el puerto 22 desde una dirección IP específica, mediante un recurso [AWS::EC2::SecurityGroup](#). Crea un volumen de Amazon EBS de 100 GB mediante un recurso [AWS::EC2::Volume](#). El volumen se crea en la misma zona de disponibilidad que la instancia, según lo especifique la función GetAtt, y se monta en la instancia del dispositivo /dev/sdh.

Para más información sobre la creación de una instantánea en EBS, consulte [Creación de un volumen de 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
```

Creación de grupos de seguridad con reglas de entrada

En los siguientes fragmentos de ejemplo, se muestra cómo configurar grupos de seguridad con reglas de entrada específicas mediante CloudFormation.

Fragmentos

- [Creación de un grupo de seguridad con reglas de entrada para el acceso SSH y HTTP](#)
- [Creación de un grupo de seguridad con reglas de entrada para el acceso SSH y HTTP a partir de rangos CIDR especificados](#)
- [Creación de grupos de seguridad de referencia cruzada con reglas de entrada](#)

Creación de un grupo de seguridad con reglas de entrada para el acceso SSH y HTTP

Este fragmento de código muestra un recurso [AWS::EC2::SecurityGroup](#) que describe dos reglas de entrada del grupo de seguridad. La primera regla de entrada permite el acceso SSH (puerto 22) desde un grupo de seguridad existente denominado MyAdminSecurityGroup, que es propiedad de la cuenta de AWS con el número de cuenta 1111-2222-3333. La segunda regla de entrada permite el acceso HTTP (puerto 80) desde un grupo de seguridad diferente denominado MySecurityGroupCreatedInCFN, que se crea en la misma plantilla. La función Ref se utiliza para hacer referencia al nombre lógico del grupo de seguridad creado en la misma plantilla.

En la primera regla de entrada, debe agregar un valor para las propiedades SourceSecurityGroupName y SourceSecurityGroupOwnerId. En la segunda regla de entrada, MySecurityGroupCreatedInCFNTemplate hace referencia a un grupo de seguridad diferente, que se crea en la misma plantilla. Compruebe que el nombre lógico MySecurityGroupCreatedInCFNTemplate coincide con el nombre lógico real del recurso del grupo de seguridad que especificó en la plantilla más grande.

Para obtener más información sobre los grupos de seguridad, consulte [Grupos de seguridad de Amazon EC2 para instancias de EC2](#) en la Guía del usuario de Amazon EC2.

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
```

Creación de un grupo de seguridad con reglas de entrada para el acceso SSH y HTTP a partir de rangos CIDR especificados

El siguiente fragmento crea un grupo de seguridad para una instancia de Amazon EC2 con dos reglas de entrada. Las reglas de entrada permiten el tráfico TCP entrante en los puertos especificados desde los rangos de CIDR designados. Se utiliza un recurso [AWS::EC2::SecurityGroup](#) para especificar las reglas. Debe especificar un protocolo para cada regla. En el caso de TCP, también debe especificar un puerto o intervalo de puertos. Si no especifica un grupo de seguridad de origen o un rango de CIDR, la pila se iniciará correctamente, pero la regla no se aplicará al grupo de seguridad.

Para obtener más información sobre los grupos de seguridad, consulte [Grupos de seguridad de Amazon EC2 para instancias de EC2](#) en la Guía del usuario de Amazon EC2.

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
```

Creación de grupos de seguridad de referencia cruzada con reglas de entrada

El siguiente fragmento de código utiliza el recurso [AWS::EC2::SecurityGroup](#) para crear dos grupos de seguridad de Amazon EC2, SGroup1 y SGroup2. Las reglas de ingreso que permiten la comunicación entre los dos grupos de seguridad se crean mediante el recurso [AWS::EC2::SecurityGroupIngress](#). SGroup1Ingress establece una regla de entrada para SGroup1 que permite el tráfico TCP entrante en el puerto 80 desde el grupo de seguridad de origen, SGroup2. SGroup2Ingress establece una regla de entrada para SGroup2 que permite el tráfico TCP entrante en el puerto 80 desde el grupo de seguridad de origen, 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
```

Creación de un equilibrador de carga elástica con una regla de ingreso de grupos de seguridad

La siguiente plantilla crea un recurso [AWS::ElasticLoadBalancing::LoadBalancer](#) en la zona de disponibilidad especificada. El recurso [AWS::ElasticLoadBalancing::LoadBalancer](#) está configurado para escuchar el tráfico HTTP en el puerto 80 y dirigir las solicitudes a las instancias también en el puerto 80. El equilibrador de carga elástico es responsable de equilibrar la carga del tráfico HTTP entrante entre las instancias.

Además, esta plantilla genera un recurso [AWS::EC2::SecurityGroup](#) asociado al equilibrador de carga. Este grupo de seguridad se crea con una única regla de entrada, descrita como `ELB ingress group`, que permite el tráfico TCP entrante en el puerto 80. El origen de esta regla de entrada se define mediante la función `Fn::GetAtt` para recuperar los atributos del recurso del equilibrador de carga. `SourceSecurityGroupOwnerId` utiliza `Fn::GetAtt` para obtener el grupo de seguridad `OwnerAlias` del equilibrador de carga. `SourceSecurityGroupName` utiliza `Fn::GetAtt` para obtener el `GroupName` del grupo de seguridad de origen del ELB.

Esta configuración garantiza una comunicación segura entre el ELB y las instancias.

Para obtener más información sobre balanceadores de carga, consulte la [Guía del usuario de 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
```

Asignación y asociación de una dirección IP elástica con CloudFormation

Los siguientes fragmentos de plantilla son ejemplos relacionados con las direcciones IP elásticas (EIP) en Amazon EC2. Estos ejemplos abarcan la asignación, la asociación y la administración de las EIP para sus instancias.

Fragmentos de código de ejemplo

- [Asignación de una dirección IP elástica con una instancia de Amazon EC2](#)
- [Asociación de una dirección IP elástica a una instancia de Amazon EC2 mediante la especificación de la dirección IP](#)
- [Asociación de una dirección IP elástica a una instancia de Amazon EC2 mediante la especificación del ID de asignación de la dirección IP](#)

Asignación de una dirección IP elástica con una instancia de Amazon EC2

En el siguiente fragmento se asigna una dirección IP elástica (EIP) de Amazon EC2 y se asocia a una instancia de Amazon EC2 mediante un recurso [AWS::EC2::EIP](#). Puede asignar una dirección IP elástica de un grupo de direcciones propiedad de AWS o de un grupo de direcciones creado a partir de un intervalo de direcciones IPv4 públicas que ha llevado a AWS para su uso con los recursos

de AWS mediante sus [propias direcciones IP \(BYOIP\)](#). En este ejemplo, la EIP se asigna desde un grupo de direcciones propiedad de AWS.

Para obtener más información acerca de las direcciones IP elásticas, consulte [Direcciones IP elásticas](#) en la Guía del usuario de Amazon EC2.

JSON

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

YAML

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

Asociación de una dirección IP elástica a una instancia de Amazon EC2 mediante la especificación de la dirección IP

En el siguiente fragmento se asigna una dirección IP elástica (EIP) de Amazon EC2 y se asocia a una instancia de EC2 mediante un recurso [AWS::EC2::EIPAssociation](#). Primero debe asignar una dirección IP elástica para utilizarla en su cuenta. Se puede asociar una dirección IP elástica a una instancia única.

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
```

Asociación de una dirección IP elástica a una instancia de Amazon EC2 mediante la especificación del ID de asignación de la dirección IP

En el siguiente fragmento se asocia una dirección IP elástica existente a una instancia de Amazon EC2 mediante la especificación del ID de asignación mediante un recurso [AWS::EC2::EIPAssociation](#). Se asigna un ID de asignación a una dirección IP elástica tras la asignación de dirección IP elástica.

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
```

Configuración de los recursos de Amazon VPC con CloudFormation

En esta sección se proporcionan ejemplos para configurar los recursos de Amazon VPC con CloudFormation. Las VPC le permiten crear una red virtual dentro de AWS, y estos fragmentos muestran cómo configurar algunos aspectos de las VPC para cumplir con sus requisitos de red.

Fragmentos de código de ejemplo

- [Habilitación del acceso a Internet solo de salida de direcciones IPv6 en una VPC](#)
- [Fragmentos de código de plantillas de la Interfaz de red elástica \(ENI\)](#)

Habilitación del acceso a Internet solo de salida de direcciones IPv6 en una VPC

Una puerta de enlace de Internet de solo salida permite que las instancias de una VPC accedan a Internet e impiden que los recursos de Internet se comuniquen con las instancias. El siguiente fragmento habilita el acceso a Internet solo de salida de direcciones IPv6 desde una VPC. Crea una VPC con un rango de direcciones IPv4 de 10.0.0/16 con un recurso [AWS::EC2::VPC](#). Se asocia una tabla de enrutamiento a este recurso de VPC mediante un recurso [AWS::EC2::RouteTable](#). La tabla de enrutamiento administra las rutas de las instancias dentro de la VPC. Se utiliza un recurso [AWS::EC2::EgressOnlyInternetGateway](#) para crear una puerta de enlace de Internet solo de salida que permita la comunicación de las direcciones IPv6 para el tráfico saliente de las instancias de la VPC y, al mismo tiempo, evite el tráfico entrante. Se especifica un recurso [AWS::EC2::Route](#) para crear una ruta IPv6 en la tabla de enrutamiento que dirija todo el tráfico IPv6 saliente (: : /0) a la puerta de enlace de Internet solo de salida.

Para obtener más información sobre las puertas de enlace de Internet de solo salida, consulte [Habilitar el tráfico IPv6 saliente mediante una puerta de enlace de Internet de solo salida](#) en la Guía del usuario de 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"
```

Fragmentos de código de plantillas de la Interfaz de red elástica (ENI)

Creación de una instancia de Amazon EC2 con interfaces de red elásticas (ENI) conectadas

En el siguiente fragmento de ejemplo se crea una instancia de Amazon EC2 con un recurso [AWS::EC2::Instance](#) en la VPC y subred de Amazon especificadas. Conecta dos interfaces de red (ENI) a la instancia, asocia direcciones IP elásticas a las instancias a través de las ENI conectadas y configura el grupo de seguridad para el acceso SSH y HTTP. Los datos de usuario se proporcionan a la instancia como parte de la configuración de lanzamiento cuando se crea la instancia. Los datos de usuario incluyen un script codificado en un formato base64 para garantizar que se transfiera a la instancia. Cuando se lanza la instancia, el script se ejecuta automáticamente como parte del proceso de arranque. Instala `ec2-net-utils`, configura las interfaces de red e inicia el servicio HTTP.

Para determinar la imagen de máquina de Amazon (AMI) adecuada en función de la región seleccionada, el fragmento utiliza una función `Fn::FindInMap` que busca valores en una asignación `RegionMap`. Esta asignación debe definirse en la plantilla más grande. Las dos interfaces de red se crean con recursos [AWS::EC2::NetworkInterface](#). Las direcciones IP elásticas se especifican mediante recursos [AWS::EC2::EIP](#) asignados al dominio `vpc`. Estas direcciones IP elásticas se asocian a las interfaces de red mediante recursos [AWS::EC2::EIPAssociation](#).

La sección `Outputs` define valores o recursos a los que desea acceder después de que se haya creado la pila. En este fragmento, el resultado definido es `InstancePublicIp`, que representa la dirección IP pública de la instancia de EC2 creada por la pila. Puede recuperar este resultado en la pestaña Salida, en la consola de CloudFormation o mediante el comando [describe-stacks](#).

Para obtener más información acerca de las interfaces de red, consulte [Interfaces de red elásticas](#).

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
```

Ejemplos de plantillas de Amazon Elastic Container Service

Amazon Elastic Container Service (Amazon ECS) es un servicio de administración de contenedores que facilita la tarea de ejecutar, detener y administrar contenedores de Docker en un clúster de instancias de Amazon Elastic Compute Cloud (Amazon EC2).

Cree un clúster con la AMI optimizada para Amazon ECS basada en AL2023

Defina un clúster que utilice un proveedor de capacidad que lance instancias de AL2023 en Amazon EC2.

Important

Para los últimos ID de la AMI, consulte la [AMI optimizada para Amazon ECS](#) en la Guía para desarrolladores de Amazon Elastic Container Service.

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "EC2 ECS cluster that starts out empty, with no EC2 instances yet. An ECS capacity provider automatically launches more EC2 instances as required on the fly when you request ECS to launch services or standalone tasks.",
  "Parameters": {
    "InstanceType": {
      "Type": "String",
      "Description": "EC2 instance type",
      "Default": "t2.medium",
      "AllowedValues": [
        "t1.micro",
        "t2.2xlarge",
        "t2.large",
        "t2.medium",
        "t2.micro",
        "t2.nano",
        "t2.small",
        "t2.xlarge",
        "t3.2xlarge",
        "t3.large",
        "t3.medium",
        "t3.micro",
        "t3.nano",
        "t3.small",
        "t3.xlarge"
      ]
    },
    "DesiredCapacity": {
      "Type": "Number",
      "Default": "0",
      "Description": "Number of EC2 instances to launch in your ECS cluster."
    },
    "MaxSize": {
      "Type": "Number",
      "Default": "100",
      "Description": "Maximum number of EC2 instances that can be launched in your ECS cluster."
    },
    "ECSAMI": {
      "Description": "The Amazon Machine Image ID used for the cluster",
      "Type": "AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>",

```

```

        "Default": "/aws/service/ecs/optimized-ami/amazon-linux-2023/recommended/
image_id"
    },
    "VpcId": {
        "Type": "AWS::EC2::VPC::Id",
        "Description": "VPC ID where the ECS cluster is launched",
        "Default": "vpc-1234567890abcdef0"
    },
    "SubnetIds": {
        "Type": "List<AWS::EC2::Subnet::Id>",
        "Description": "List of subnet IDs where the EC2 instances will be launched",
        "Default": "subnet-021345abcdef67890"
    }
},
"Resources": {
    "ECSCluster": {
        "Type": "AWS::ECS::Cluster",
        "Properties": {
            "ClusterSettings": [
                {
                    "Name": "containerInsights",
                    "Value": "enabled"
                }
            ]
        }
    },
    "ECSAutoScalingGroup": {
        "Type": "AWS::AutoScaling::AutoScalingGroup",
        "DependsOn": [
            "ECSCluster",
            "EC2Role"
        ],
        "Properties": {
            "VPCZoneIdentifier": {
                "Ref": "SubnetIds"
            },
            "LaunchTemplate": {
                "LaunchTemplateId": {
                    "Ref": "ContainerInstances"
                },
                "Version": {
                    "Fn::GetAtt": [
                        "ContainerInstances",
                        "LatestVersionNumber"
                    ]
                }
            }
        }
    }
}

```



```

    ],
    "MinSize": 0,
    "MaxSize": {
      "Ref": "MaxSize"
    },
    "DesiredCapacity": {
      "Ref": "DesiredCapacity"
    },
    "NewInstancesProtectedFromScaleIn": true
  },
  "UpdatePolicy": {
    "AutoScalingReplacingUpdate": {
      "WillReplace": "true"
    }
  }
},
"ContainerInstances": {
  "Type": "AWS::EC2::LaunchTemplate",
  "Properties": {
    "LaunchTemplateName": "asg-launch-template",
    "LaunchTemplateData": {
      "ImageId": {
        "Ref": "ECSAMI"
      },
      "InstanceType": {
        "Ref": "InstanceType"
      },
      "IamInstanceProfile": {
        "Name": {
          "Ref": "EC2InstanceProfile"
        }
      },
      "SecurityGroupIds": [
        {
          "Ref": "ContainerHostSecurityGroup"
        }
      ],
      "UserData": {
        "Fn::Base64": {
          "Fn::Sub": "#!/bin/bash -xe\n echo ECS_CLUSTER=${ECSCluster}
>> /etc/ecs/ecs.config\n yum install -y aws-cfn-bootstrap\n /opt/aws/bin/cfn-init -

```

```

v --stack ${AWS::StackId} --resource ContainerInstances --configsets full_install --
region ${AWS::Region} &\n"
    }
    },
    "MetadataOptions": {
        "HttpEndpoint": "enabled",
        "HttpTokens": "required"
    }
}
},
"EC2InstanceProfile": {
    "Type": "AWS::IAM::InstanceProfile",
    "Properties": {
        "Path": "/",
        "Roles": [
            {
                "Ref": "EC2Role"
            }
        ]
    }
},
"CapacityProvider": {
    "Type": "AWS::ECS::CapacityProvider",
    "Properties": {
        "AutoScalingGroupProvider": {
            "AutoScalingGroupArn": {
                "Ref": "ECSAutoScalingGroup"
            }
        },
        "ManagedScaling": {
            "InstanceWarmupPeriod": 60,
            "MinimumScalingStepSize": 1,
            "MaximumScalingStepSize": 100,
            "Status": "ENABLED",
            "TargetCapacity": 100
        },
        "ManagedTerminationProtection": "ENABLED"
    }
}
},
"CapacityProviderAssociation": {
    "Type": "AWS::ECS::ClusterCapacityProviderAssociations",
    "Properties": {
        "CapacityProviders": [

```

```

        {
            "Ref": "CapacityProvider"
        }
    ],
    "Cluster": {
        "Ref": "ECSCluster"
    },
    "DefaultCapacityProviderStrategy": [
        {
            "Base": 0,
            "CapacityProvider": {
                "Ref": "CapacityProvider"
            },
            "Weight": 1
        }
    ]
},
"ContainerHostSecurityGroup": {
    "Type": "AWS::EC2::SecurityGroup",
    "Properties": {
        "GroupDescription": "Access to the EC2 hosts that run containers",
        "VpcId": {
            "Ref": "VpcId"
        }
    }
},
"EC2Role": {
    "Type": "AWS::IAM::Role",
    "Properties": {
        "AssumeRolePolicyDocument": {
            "Statement": [
                {
                    "Effect": "Allow",
                    "Principal": {
                        "Service": [
                            "ec2.amazonaws.com"
                        ]
                    },
                    "Action": [
                        "sts:AssumeRole"
                    ]
                }
            ]
        }
    }
}
]

```

```

    },
    "Path": "/",
    "ManagedPolicyArns": [
        "arn:aws:iam::aws:policy/service-role/
AmazonEC2ContainerServiceforEC2Role",
        "arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore"
    ]
},
"ECSTaskExecutionRole": {
    "Type": "AWS::IAM::Role",
    "Properties": {
        "AssumeRolePolicyDocument": {
            "Statement": [
                {
                    "Effect": "Allow",
                    "Principal": {
                        "Service": [
                            "ecs-tasks.amazonaws.com"
                        ]
                    },
                    "Action": [
                        "sts:AssumeRole"
                    ],
                    "Condition": {
                        "ArnLike": {
                            "aws:SourceArn": {
                                "Fn::Sub": "arn:${AWS::Partition}:ecs:
${AWS::Region}:${AWS::AccountId}:*"
                            }
                        },
                        "StringEquals": {
                            "aws:SourceAccount": {
                                "Fn::Sub": "${AWS::AccountId}"
                            }
                        }
                    }
                }
            ]
        }
    }
},
    "Path": "/",
    "ManagedPolicyArns": [
        "arn:aws:iam::aws:policy/service-role/
AmazonECSTaskExecutionRolePolicy"

```

```

        ]
    }
}
},
"Outputs": {
    "ClusterName": {
        "Description": "The ECS cluster into which to launch resources",
        "Value": "ECSCluster"
    },
    "ECSTaskExecutionRole": {
        "Description": "The role used to start up a task",
        "Value": "ECSTaskExecutionRole"
    },
    "CapacityProvider": {
        "Description": "The cluster capacity provider that the service should use to
request capacity when it wants to start up a task",
        "Value": "CapacityProvider"
    }
}
}
}

```

YAML

AWSTemplateFormatVersion: 2010-09-09

Description: EC2 ECS cluster that starts out empty, with no EC2 instances yet.

An ECS capacity provider automatically launches more EC2 instances as required on the fly when you request ECS to launch services or standalone tasks.

Parameters:

InstanceType:

Type: String

Description: EC2 instance type

Default: "t2.medium"

AllowedValues:

- t1.micro
- t2.2xlarge
- t2.large
- t2.medium
- t2.micro
- t2.nano
- t2.small
- t2.xlarge
- t3.2xlarge

- t3.large
- t3.medium
- t3.micro
- t3.nano
- t3.small
- t3.xlarge

DesiredCapacity:

Type: Number

Default: "0"

Description: Number of EC2 instances to launch in your ECS cluster.

MaxSize:

Type: Number

Default: "100"

Description: Maximum number of EC2 instances that can be launched in your ECS cluster.

ECSAMI:

Description: The Amazon Machine Image ID used for the cluster

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

Default: /aws/service/ecs/optimized-ami/amazon-linux-2023/recommended/image_id

VpcId:

Type: AWS::EC2::VPC::Id

Description: VPC ID where the ECS cluster is launched

Default: vpc-1234567890abcdef0

SubnetIds:

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

Description: List of subnet IDs where the EC2 instances will be launched

Default: "subnet-021345abcdef67890"

Resources:

```
# This is authorizes ECS to manage resources on your
# account on your behalf. This role is likely already created on your account
# ECSRole:
#   Type: AWS::IAM::ServiceLinkedRole
#   Properties:
#     AWSServiceName: 'ecs.amazonaws.com'
```

ECS Resources**ECScluster:**

Type: AWS::ECS::Cluster

Properties:**ClusterSettings:**

- Name: containerInsights
- Value: enabled

```
# Autoscaling group. This launches the actual EC2 instances that will register
```

```

# themselves as members of the cluster, and run the docker containers.
ECSAutoScalingGroup:
  Type: AWS::AutoScaling::AutoScalingGroup
  DependsOn:
    # This is to ensure that the ASG gets deleted first before these
    # resources, when it comes to stack teardown.
    - ECSCluster
    - EC2Role
  Properties:
    VPCZoneIdentifier:
      Ref: SubnetIds
    LaunchTemplate:
      LaunchTemplateId: !Ref ContainerInstances
      Version: !GetAtt ContainerInstances.LatestVersionNumber
    MinSize: 0
    MaxSize:
      Ref: MaxSize
    DesiredCapacity:
      Ref: DesiredCapacity
    NewInstancesProtectedFromScaleIn: true
  UpdatePolicy:
    AutoScalingReplacingUpdate:
      WillReplace: "true"
# The config for each instance that is added to the cluster
ContainerInstances:
  Type: AWS::EC2::LaunchTemplate
  Properties:
    LaunchTemplateName: "asg-launch-template"
    LaunchTemplateData:
      ImageId:
        Ref: ECSAMI
      InstanceType:
        Ref: InstanceType
      IamInstanceProfile:
        Name: !Ref EC2InstanceProfile
      SecurityGroupIds:
        - !Ref ContainerHostSecurityGroup
      # This injected configuration file is how the EC2 instance
      # knows which ECS cluster on your AWS account it should be joining
      UserData:
        Fn::Base64: !Sub |
          #!/bin/bash -xe
          echo ECS_CLUSTER=${ECSCluster} >> /etc/ecs/ecs.config
          yum install -y aws-cfn-bootstrap

```

```

    /opt/aws/bin/cfn-init -v --stack ${AWS::StackId} --resource
ContainerInstances --configsets full_install --region ${AWS::Region} &
    # Disable IMDSv1, and require IMDSv2
    MetadataOptions:
        HttpEndpoint: enabled
        HttpTokens: required
EC2InstanceProfile:
    Type: AWS::IAM::InstanceProfile
    Properties:
        Path: /
        Roles:
            - !Ref EC2Role
# Create an ECS capacity provider to attach the ASG to the ECS cluster
# so that it autoscales as we launch more containers
CapacityProvider:
    Type: AWS::ECS::CapacityProvider
    Properties:
        AutoScalingGroupProvider:
            AutoScalingGroupArn: !Ref ECSAutoScalingGroup
        ManagedScaling:
            InstanceWarmupPeriod: 60
            MinimumScalingStepSize: 1
            MaximumScalingStepSize: 100
            Status: ENABLED
            # Percentage of cluster reservation to try to maintain
            TargetCapacity: 100
        ManagedTerminationProtection: ENABLED
# Create a cluster capacity provider association so that the cluster
# will use the capacity provider
CapacityProviderAssociation:
    Type: AWS::ECS::ClusterCapacityProviderAssociations
    Properties:
        CapacityProviders:
            - !Ref CapacityProvider
        Cluster: !Ref ECSCluster
        DefaultCapacityProviderStrategy:
            - Base: 0
              CapacityProvider: !Ref CapacityProvider
              Weight: 1
# A security group for the EC2 hosts that will run the containers.
# This can be used to limit incoming traffic to or outgoing traffic
# from the container's host EC2 instance.
ContainerHostSecurityGroup:
    Type: AWS::EC2::SecurityGroup

```



```

Properties:
  GroupDescription: Access to the EC2 hosts that run containers
  VpcId:
    Ref: VpcId
# Role for the EC2 hosts. This allows the ECS agent on the EC2 hosts
# to communicate with the ECS control plane, as well as download the docker
# images from ECR to run on your host.
EC2Role:
  Type: AWS::IAM::Role
  Properties:
    AssumeRolePolicyDocument:
      Statement:
        - Effect: Allow
          Principal:
            Service:
              - ec2.amazonaws.com
          Action:
            - sts:AssumeRole
    Path: /
    ManagedPolicyArns:
      # See reference: https://docs.aws.amazon.com/AmazonECS/latest/developerguide/
security-iam-awsmanpol.html#security-iam-awsmanpol-AmazonEC2ContainerServiceforEC2Role
      - arn:aws:iam::aws:policy/service-role/AmazonEC2ContainerServiceforEC2Role
      # This managed policy allows us to connect to the instance using SSM
      - arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore
# This is a role which is used within Fargate to allow the Fargate agent
# to download images, and upload logs.
ECSTaskExecutionRole:
  Type: AWS::IAM::Role
  Properties:
    AssumeRolePolicyDocument:
      Statement:
        - Effect: Allow
          Principal:
            Service:
              - ecs-tasks.amazonaws.com
          Action:
            - sts:AssumeRole
      Condition:
        ArnLike:
          aws:SourceArn: !Sub arn:${AWS::Partition}:ecs:${AWS::Region}:
${AWS::AccountId}:*
        StringEquals:
          aws:SourceAccount: !Sub ${AWS::AccountId}

```

```

    Path: /
    # This role enables all features of ECS. See reference:
    # https://docs.aws.amazon.com/AmazonECS/latest/developerguide/security-iam-awsmanpol.html#security-iam-awsmanpol-AmazonECSTaskExecutionRolePolicy
    ManagedPolicyArns:
      - arn:aws:iam::aws:policy/service-role/AmazonECSTaskExecutionRolePolicy
Outputs:
  ClusterName:
    Description: The ECS cluster into which to launch resources
    Value: ECSCluster
  ECSTaskExecutionRole:
    Description: The role used to start up a task
    Value: ECSTaskExecutionRole
  CapacityProvider:
    Description: The cluster capacity provider that the service should use to
      request capacity when it wants to start up a task
    Value: CapacityProvider

```

Implementación de un servicio

La siguiente plantilla define un servicio que utiliza el proveedor de capacidad para solicitar la capacidad de ejecución de AL2023. Los contenedores se incorporarán a las instancias de AL2023 a medida que se encuentren en línea:

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

```

Plantilla de ejemplo de Amazon Elastic File System

Amazon Elastic File System (Amazon EFS) es un servicio de almacenamiento de archivos para instancias Amazon Elastic Compute Cloud (Amazon EC2). Con Amazon EFS, sus aplicaciones cuentan con almacenamiento cuando lo necesitan, ya que la capacidad de almacenamiento crece y se reduce automáticamente a medida que añade o elimina archivos.

La siguiente plantilla de ejemplo implementa instancias de EC2 (en un grupo de Auto Scaling) que están asociadas con un sistema de archivos de Amazon EFS. Para asociar las instancias con el sistema de archivos, las instancias ejecutan el script auxiliar `cfn-init` que descarga e instala el paquete `yum nfs-utils`, crea un nuevo directorio y, a continuación, utiliza el nombre de DNS del sistema de archivos para montar el sistema de archivos en ese directorio. El nombre de DNS del sistema de archivos se resuelve en una dirección IP del destino de montaje en la zona de disponibilidad de la instancia Amazon EC2. Para obtener más información sobre la estructura de nombres de DNS, consulte [Montaje de sistemas de archivos](#) en la Guía del usuario de Amazon Elastic File System.

Para medir la actividad del Sistema de archivos de red (NFS), la plantilla incluye métricas de Amazon CloudWatch personalizadas. La plantilla también crea una VPC, una subred y grupos de seguridad. Para permitir que las instancias se comuniquen con el sistema de archivos, la VPC debe tener el DNS habilitado, y el destino de montaje y las instancias de EC2 tienen que estar en la misma zona de disponibilidad (AZ), especificada por la subred.

El grupo de seguridad del destino de montaje permite una conexión de red al puerto TCP 2049, que es necesaria para que un cliente NFSv4 monte un sistema de archivos. Para obtener más información acerca de los grupos de seguridad para las instancias de EC2 y los destinos de montaje, consulte [Seguridad](#) en la [Guía del usuario de Amazon Elastic File System](#).

Note

Si realiza una actualización en el destino de montaje que provoca que haya que sustituirlo, las instancias o aplicaciones que utilizan el sistema de archivos asociados puede verse afectado. Esto puede provocar la pérdida de escrituras sin confirmar. Para evitar interrupciones, detenga las instancias al actualizar el destino de montaje configurando la capacidad deseada en cero. Esto permite que las instancias desmonten el sistema de archivos antes de que se elimine el destino de montaje. Una vez completada la actualización de montaje, comience las instancias en una actualización posterior configurando la capacidad deseada.

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 $line | tr -s ' ' '\n`,\n",
        "        AWS_COMMAND=\"aws cloudwatch put-metric-data --region ",
{ "Ref": "AWS::Region" }, "\n",
        "        MOD=$(( 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-
data/instance-id)\n",
        "                CW_JSON_METRIC=\"${CW_JSON_METRIC} { \\\"MetricName\\\": \\\"
\\\"$METRIC_NAME\\\", \\\"Dimensions\\\": [{\\\"Name\\\": \\\"InstanceId\\\", \\\"Value\\\"
\\\": \\\"$INSTANCE_ID\\\"} ], \\\"Value\\\": $METRIC_VALUE },\n",
        "                unset METRIC_NAME\n",
        "                unset METRIC_VALUE\n",
        "\n",
        "                METRIC_COUNTER=$((METRIC_COUNTER+1))\n",
        "                if [ $METRIC_COUNTER -eq 20 ]; then\n",
        "                    # 20 is max metric collection size, so we have to
submit here\n",
        "                    aws cloudwatch put-metric-data --region ", { "Ref":
"AWS::Region" }, " --cli-input-json \"`echo $CW_JSON_OPEN ${CW_JSON_METRIC%?}
$CW_JSON_CLOSE`\n",
        "\n",
        "                    # reset\n",
        "                    METRIC_COUNTER=0\n",
        "                    CW_JSON_METRIC=''\n",
        "                fi\n",
        "            fi\n",
        "\n",
        "\n",
        "\n",
        "        COUNTER=$((COUNTER+1))\n",
        "    fi\n",
        "\n",
        "    if [[ \"$line\" == \"Client nfs v4:\" ]]; then\n",

```

```

        "      # the next line is the good stuff \n",
        "      COUNTER=$((COUNTER+1))\n",
        "      fi\n",
        "      done <<< \"\${INPUT}\n\n",
        "done\n",
        "\n",
        "# submit whatever is left\n",
        "aws cloudwatch put-metric-data --region ", { "Ref":
"AWS::Region" }, " --cli-input-json \"`echo $CW_JSON_OPEN ${CW_JSON_METRIC%?}
$CW_JSON_CLOSE`\n"
    ] ] },
    "mode": "000755",
    "owner": "ec2-user",
    "group": "ec2-user"
  },
  "/home/ec2-user/crontab" : {
    "content" : { "Fn::Join" : [ "", [
      "** * * * * /usr/sbin/nfsstat | /home/ec2-user/post_nfsstat\n"
    ] ] },
    "owner": "ec2-user",
    "group": "ec2-user"
  }
},
"commands" : {
  "01_createdir" : {
    "command" : {"Fn::Join" : [ "", [ "mkdir /", { "Ref" :
"MountPoint" } ] ] }
  }
},
"mount" : {
  "commands" : {
    "01_mount" : {
      "command" : { "Fn::Sub": "sudo mount -t nfs4 -o
nfsvers=4.1,rsize=1048576,wsiz=1048576,hard,timeo=600,retrans=2 ${FileSystem}.efs.
${AWS::Region}.amazonaws.com:/ /${MountPoint}" }
    },
    "02_permissions" : {
      "command" : {"Fn::Join" : [ "", [ "chown ec2-user:ec2-user /",
{ "Ref" : "MountPoint" } ] ] }
    }
  }
}
}
}

```



```

    },
    "Properties": {
      "AssociatePublicIpAddress" : true,
      "ImageId": {
        "Fn::FindInMap": [ "AWSRegionArch2AMI", { "Ref": "AWS::Region" }, {
          "Fn::FindInMap": [ "AWSInstanceType2Arch", { "Ref": "InstanceType" },
"Arch" ]
        } ]
      },
      "InstanceType": { "Ref": "InstanceType" },
      "KeyName": { "Ref": "KeyName" },
      "SecurityGroups": [ { "Ref": "InstanceSecurityGroup" } ],
      "IamInstanceProfile" : { "Ref" : "CloudWatchPutMetricsInstanceProfile" },
      "UserData" : { "Fn::Base64" : { "Fn::Join" : [ "", [
        "#!/bin/bash -xe\n",
        "yum install -y aws-cfn-bootstrap\n",

        "/opt/aws/bin/cfn-init -v ",
        "      --stack ", { "Ref" : "AWS::StackName" },
        "      --resource LaunchConfiguration ",
        "      --configsets MountConfig ",
        "      --region ", { "Ref" : "AWS::Region" }, "\n",

        "crontab /home/ec2-user/crontab\n",

        "/opt/aws/bin/cfn-signal -e $? ",
        "      --stack ", { "Ref" : "AWS::StackName" },
        "      --resource AutoScalingGroup ",
        "      --region ", { "Ref" : "AWS::Region" }, "\n"
      ] ] ] }
    ] ] ] }
  },
  "AutoScalingGroup": {
    "Type": "AWS::AutoScaling::AutoScalingGroup",
    "DependsOn": ["MountTarget", "GatewayToInternet"],
    "CreationPolicy" : {
      "ResourceSignal" : {
        "Timeout" : "PT15M",
        "Count" : { "Ref": "AsgMaxSize" }
      }
    },
  },
  "Properties": {
    "VPCZoneIdentifier": [ { "Ref": "Subnet" } ],
    "LaunchConfigurationName": { "Ref": "LaunchConfiguration" },

```

```

        "MinSize": "1",
        "MaxSize": { "Ref": "AsgMaxSize" },
        "DesiredCapacity": { "Ref": "AsgMaxSize" },
        "Tags": [ {
            "Key": "Name",
            "Value": "EFS FileSystem Mounted Instance",
            "PropagateAtLaunch": "true"
        } ]
    }
}
},
"Outputs" : {
    "MountTargetID" : {
        "Description" : "Mount target ID",
        "Value" : { "Ref" : "MountTarget" }
    },
    "FileSystemID" : {
        "Description" : "File system ID",
        "Value" : { "Ref" : "FileSystem" }
    }
}
}
}

```

YAML

AWSTemplateFormatVersion: '2010-09-09'

Description: This template creates an Amazon EFS file system and mount target and associates it with Amazon EC2 instances in an Auto Scaling group. ****WARNING**** This template creates Amazon EC2 instances and related resources. You will be billed for the AWS resources used if you create a stack from this template.

Parameters:

InstanceType:

Description: WebServer EC2 instance type

Type: String

Default: t2.small

AllowedValues:

- t1.micro
- t2.nano
- t2.micro
- t2.small
- t2.medium
- t2.large
- m1.small

- m1.medium
- m1.large
- m1.xlarge
- m2.xlarge
- m2.2xlarge
- m2.4xlarge
- m3.medium
- m3.large
- m3.xlarge
- m3.2xlarge
- m4.large
- m4.xlarge
- m4.2xlarge
- m4.4xlarge
- m4.10xlarge
- c1.medium
- c1.xlarge
- c3.large
- c3.xlarge
- c3.2xlarge
- c3.4xlarge
- c3.8xlarge
- c4.large
- c4.xlarge
- c4.2xlarge
- c4.4xlarge
- c4.8xlarge
- g2.2xlarge
- g2.8xlarge
- r3.large
- r3.xlarge
- r3.2xlarge
- r3.4xlarge
- r3.8xlarge
- i2.xlarge
- i2.2xlarge
- i2.4xlarge
- i2.8xlarge
- d2.xlarge
- d2.2xlarge
- d2.4xlarge
- d2.8xlarge
- hi1.4xlarge
- hs1.8xlarge

- cr1.8xlarge
- cc2.8xlarge
- cg1.4xlarge

ConstraintDescription: must be a valid EC2 instance type.

KeyName:

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

Description: Name of an existing EC2 key pair to enable SSH access to the ECS instances

AsgMaxSize:

Type: Number

Description: Maximum size and initial desired capacity of Auto Scaling Group

Default: '2'

SSHLocation:

Description: The IP address range that can be used to connect to the EC2 instances by using SSH

Type: String

MinLength: '9'

MaxLength: '18'

Default: 0.0.0.0/0

AllowedPattern: "(\\d{1,3})\\. (\\d{1,3})\\. (\\d{1,3})\\. (\\d{1,3})/(\\d{1,2})"

ConstraintDescription: must be a valid IP CIDR range of the form x.x.x.x/x.

VolumeName:

Description: The name to be used for the EFS volume

Type: String

MinLength: '1'

Default: myEFSvolume

MountPoint:

Description: The Linux mount point for the EFS volume

Type: String

MinLength: '1'

Default: myEFSvolume

Mappings:

AWSInstanceType2Arch:

```
t1.micro:
```

Arch: HVM64

t2.nano:

Arch: HVM64

```
t2.micro:
```

Arch: HVM64

```
t2.small:
```

Arch: HVM64

t2.medium:

Arch: HVM64

t2.large:

```
    Arch: HVM64
m1.small:
    Arch: HVM64
m1.medium:
    Arch: HVM64
m1.large:
    Arch: HVM64
m1.xlarge:
    Arch: HVM64
m2.xlarge:
    Arch: HVM64
m2.2xlarge:
    Arch: HVM64
m2.4xlarge:
    Arch: HVM64
m3.medium:
    Arch: HVM64
m3.large:
    Arch: HVM64
m3.xlarge:
    Arch: HVM64
m3.2xlarge:
    Arch: HVM64
m4.large:
    Arch: HVM64
m4.xlarge:
    Arch: HVM64
m4.2xlarge:
    Arch: HVM64
m4.4xlarge:
    Arch: HVM64
m4.10xlarge:
    Arch: HVM64
c1.medium:
    Arch: HVM64
c1.xlarge:
    Arch: HVM64
c3.large:
    Arch: HVM64
c3.xlarge:
    Arch: HVM64
c3.2xlarge:
    Arch: HVM64
c3.4xlarge:
```

```
    Arch: HVM64
c3.8xlarge:
    Arch: HVM64
c4.large:
    Arch: HVM64
c4.xlarge:
    Arch: HVM64
c4.2xlarge:
    Arch: HVM64
c4.4xlarge:
    Arch: HVM64
c4.8xlarge:
    Arch: HVM64
g2.2xlarge:
    Arch: HVMG2
g2.8xlarge:
    Arch: HVMG2
r3.large:
    Arch: HVM64
r3.xlarge:
    Arch: HVM64
r3.2xlarge:
    Arch: HVM64
r3.4xlarge:
    Arch: HVM64
r3.8xlarge:
    Arch: HVM64
i2.xlarge:
    Arch: HVM64
i2.2xlarge:
    Arch: HVM64
i2.4xlarge:
    Arch: HVM64
i2.8xlarge:
    Arch: HVM64
d2.xlarge:
    Arch: HVM64
d2.2xlarge:
    Arch: HVM64
d2.4xlarge:
    Arch: HVM64
d2.8xlarge:
    Arch: HVM64
hi1.4xlarge:
```

```
    Arch: HVM64
  hs1.8xlarge:
    Arch: HVM64
  cr1.8xlarge:
    Arch: HVM64
  cc2.8xlarge:
    Arch: HVM64
AWSRegionArch2AMI:
  us-east-1:
    HVM64: ami-0ff8a91507f77f867
    HVMG2: ami-0a584ac55a7631c0c
  us-west-2:
    HVM64: ami-a0cfeed8
    HVMG2: ami-0e09505bc235aa82d
  us-west-1:
    HVM64: ami-0bdb828fd58c52235
    HVMG2: ami-066ee5fd4a9ef77f1
  eu-west-1:
    HVM64: ami-047bb4163c506cd98
    HVMG2: ami-0a7c483d527806435
  eu-west-2:
    HVM64: ami-f976839e
    HVMG2: NOT_SUPPORTED
  eu-west-3:
    HVM64: ami-0ebc281c20e89ba4b
    HVMG2: NOT_SUPPORTED
  eu-central-1:
    HVM64: ami-0233214e13e500f77
    HVMG2: ami-06223d46a6d0661c7
  ap-northeast-1:
    HVM64: ami-06cd52961ce9f0d85
    HVMG2: ami-053cdd503598e4a9d
  ap-northeast-2:
    HVM64: ami-0a10b2721688ce9d2
    HVMG2: NOT_SUPPORTED
  ap-northeast-3:
    HVM64: ami-0d98120a9fb693f07
    HVMG2: NOT_SUPPORTED
  ap-southeast-1:
    HVM64: ami-08569b978cc4dfa10
    HVMG2: ami-0be9df32ae9f92309
  ap-southeast-2:
    HVM64: ami-09b42976632b27e9b
    HVMG2: ami-0a9ce9fecc3d1daf8
```

```
ap-south-1:
  HVM64: ami-0912f71e06545ad88
  HVMG2: ami-097b15e89dbdcfcf4
us-east-2:
  HVM64: ami-0b59bfac6be064b78
  HVMG2: NOT_SUPPORTED
ca-central-1:
  HVM64: ami-0b18956f
  HVMG2: NOT_SUPPORTED
sa-east-1:
  HVM64: ami-07b14488da8ea02a0
  HVMG2: NOT_SUPPORTED
cn-north-1:
  HVM64: ami-0a4eaf6c4454eda75
  HVMG2: NOT_SUPPORTED
cn-northwest-1:
  HVM64: ami-6b6a7d09
  HVMG2: NOT_SUPPORTED
```

Resources:

CloudWatchPutMetricsRole:

Type: AWS::IAM::Role

Properties:

AssumeRolePolicyDocument:

Statement:

- Effect: Allow

Principal:

Service:

- ec2.amazonaws.com

Action:

- sts:AssumeRole

Path: "/"

CloudWatchPutMetricsRolePolicy:

Type: AWS::IAM::Policy

Properties:

PolicyName: CloudWatch_PutMetricData

PolicyDocument:

Version: '2012-10-17'

Statement:

- Sid: CloudWatchPutMetricData

Effect: Allow

Action:

- cloudwatch:PutMetricData

Resource:

- "*"


```
    Roles:
      - Ref: CloudWatchPutMetricsRole
CloudWatchPutMetricsInstanceProfile:
  Type: AWS::IAM::InstanceProfile
  Properties:
    Path: "/"
    Roles:
      - Ref: CloudWatchPutMetricsRole
VPC:
  Type: AWS::EC2::VPC
  Properties:
    EnableDnsSupport: 'true'
    EnableDnsHostnames: 'true'
    CidrBlock: 10.0.0.0/16
    Tags:
      - Key: Application
        Value:
          Ref: AWS::StackId
InternetGateway:
  Type: AWS::EC2::InternetGateway
  Properties:
    Tags:
      - Key: Application
        Value:
          Ref: AWS::StackName
      - Key: Network
        Value: Public
GatewayToInternet:
  Type: AWS::EC2::VPCEGatewayAttachment
  Properties:
    VpcId:
      Ref: VPC
    InternetGatewayId:
      Ref: InternetGateway
RouteTable:
  Type: AWS::EC2::RouteTable
  Properties:
    VpcId:
      Ref: VPC
SubnetRouteTableAssoc:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Properties:
    RouteTableId:
      Ref: RouteTable
```

```
    SubnetId:
      Ref: Subnet
InternetGatewayRoute:
  Type: AWS::EC2::Route
  Properties:
    DestinationCidrBlock: 0.0.0.0/0
    RouteTableId:
      Ref: RouteTable
    GatewayId:
      Ref: InternetGateway
Subnet:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId:
      Ref: VPC
    CidrBlock: 10.0.0.0/24
    Tags:
      - Key: Application
        Value:
          Ref: AWS::StackId
InstanceSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    VpcId:
      Ref: VPC
    GroupDescription: Enable SSH access via port 22
    SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: 22
        ToPort: 22
        CidrIp:
          Ref: SSHLocation
      - IpProtocol: tcp
        FromPort: 80
        ToPort: 80
        CidrIp: 0.0.0.0/0
MountTargetSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    VpcId:
      Ref: VPC
    GroupDescription: Security group for mount target
    SecurityGroupIngress:
      - IpProtocol: tcp
```

```

    FromPort: 2049
    ToPort: 2049
    CidrIp: 0.0.0.0/0
FileSystem:
  Type: AWS::EFS::FileSystem
  Properties:
    PerformanceMode: generalPurpose
    FileSystemTags:
      - Key: Name
        Value:
          Ref: VolumeName
MountTarget:
  Type: AWS::EFS::MountTarget
  Properties:
    FileSystemId:
      Ref: FileSystem
    SubnetId:
      Ref: Subnet
    SecurityGroups:
      - Ref: MountTargetSecurityGroup
LaunchConfiguration:
  Type: AWS::AutoScaling::LaunchConfiguration
  Metadata:
    AWS::CloudFormation::Init:
      configSets:
        MountConfig:
          - setup
          - mount
      setup:
        packages:
          yum:
            nfs-utils: []
        files:
          "/home/ec2-user/post_nfsstat":
            content: !Sub |
              #!/bin/bash

              INPUT="$(cat)"
              CW_JSON_OPEN='{ "Namespace": "EFS", "MetricData": [ '
              CW_JSON_CLOSE=' ] }'
              CW_JSON_METRIC=''
              METRIC_COUNTER=0

              for COL in 1 2 3 4 5 6; do

```

```

    COUNTER=0
    METRIC_FIELD=$COL
    DATA_FIELD=$(( $COL + ( $COL - 1 ) ))

    while read line; do
        if [[ COUNTER -gt 0 ]]; then

            LINE=`echo $line | tr -s ' ' `
            AWS_COMMAND="aws cloudwatch put-metric-data --region
${AWS::Region}"

            MOD=$(( $COUNTER % 2 ))

            if [ $MOD -eq 1 ]; then
                METRIC_NAME=`echo $LINE | cut -d ' ' -f $METRIC_FIELD`
            else
                METRIC_VALUE=`echo $LINE | cut -d ' ' -f $DATA_FIELD`
            fi

            if [[ -n "$METRIC_NAME" && -n "$METRIC_VALUE" ]]; then
                INSTANCE_ID=$(curl -s http://169.254.169.254/latest/meta-data/
instance-id)

                CW_JSON_METRIC="$CW_JSON_METRIC { \"MetricName\": \"$METRIC_NAME
\", \"Dimensions\": [{\"Name\": \"InstanceId\", \"Value\": \"$INSTANCE_ID\"} ], \"Value
\": $METRIC_VALUE },"

                unset METRIC_NAME
                unset METRIC_VALUE

                METRIC_COUNTER=$((METRIC_COUNTER+1))
                if [ $METRIC_COUNTER -eq 20 ]; then
                    # 20 is max metric collection size, so we have to submit here
                    aws cloudwatch put-metric-data --region ${AWS::Region} --cli-
input-json "`echo $CW_JSON_OPEN ${!CW_JSON_METRIC%?} $CW_JSON_CLOSE`"

                    # reset
                    METRIC_COUNTER=0
                    CW_JSON_METRIC=''
                fi
            fi

            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

Fragmentos de código de plantillas de Elastic Beanstalk

Con Elastic Beanstalk puede implementar y administrar rápidamente aplicaciones en AWS sin tener que preocuparse por la infraestructura en la que se ejecutan esas aplicaciones. La siguiente plantilla de ejemplo puede ayudarle a describir los recursos de Elastic Beanstalk en su plantilla de CloudFormation.

PHP de ejemplo de Elastic Beanstalk

La siguiente plantilla de ejemplo implementa una aplicación web PHP de ejemplo almacenada en un bucket de Amazon S3. El entorno también es un entorno de escalado automático y de equilibrio de carga, con un mínimo de dos instancias de Amazon EC2 y un máximo de seis. Muestra un entorno de Elastic Beanstalk que utiliza una configuración de lanzamiento heredada. Para obtener información sobre cómo utilizar una plantilla de lanzamiento en su lugar, consulte [Plantillas de lanzamiento en la](#) Guía para desarrolladores de AWS Elastic Beanstalk.

Reemplace *solution-stack* por un nombre de pila de soluciones (versión de plataforma). Para obtener una lista de las pilas de soluciones disponibles, utilice el comando de la AWS CLI `aws elasticbeanstalk list-available-solution-stacks`.

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Resources": {
    "sampleApplication": {
      "Type": "AWS::ElasticBeanstalk::Application",
      "Properties": {
        "Description": "AWS Elastic Beanstalk Sample Application"
      }
    },
    "sampleApplicationVersion": {
      "Type": "AWS::ElasticBeanstalk::ApplicationVersion",
      "Properties": {
        "ApplicationName": {
          "Ref": "sampleApplication"
        },
        "Description": "AWS ElasticBeanstalk Sample Application Version",
        "SourceBundle": {
          "S3Bucket": {
```

```

        "Fn::Sub": "elasticbeanstalk-samples-${AWS::Region}"
    },
    "S3Key": "php-newsample-app.zip"
}
},
"sampleConfigurationTemplate": {
    "Type": "AWS::ElasticBeanstalk::ConfigurationTemplate",
    "Properties": {
        "ApplicationName": {
            "Ref": "sampleApplication"
        },
        "Description": "AWS ElasticBeanstalk Sample Configuration Template",
        "OptionSettings": [
            {
                "Namespace": "aws:autoscaling:asg",
                "OptionName": "MinSize",
                "Value": "2"
            },
            {
                "Namespace": "aws:autoscaling:asg",
                "OptionName": "MaxSize",
                "Value": "6"
            },
            {
                "Namespace": "aws:elasticbeanstalk:environment",
                "OptionName": "EnvironmentType",
                "Value": "LoadBalanced"
            },
            {
                "Namespace": "aws:autoscaling:launchconfiguration",
                "OptionName": "IamInstanceProfile",
                "Value": {
                    "Ref": "MyInstanceProfile"
                }
            }
        ],
        "SolutionStackName": "solution-stack"
    }
},
"sampleEnvironment": {
    "Type": "AWS::ElasticBeanstalk::Environment",
    "Properties": {
        "ApplicationName": {

```



```

        "Ref": "sampleApplication"
    },
    "Description": "AWS ElasticBeanstalk Sample Environment",
    "TemplateName": {
        "Ref": "sampleConfigurationTemplate"
    },
    "VersionLabel": {
        "Ref": "sampleApplicationVersion"
    }
},
"MyInstanceRole": {
    "Type": "AWS::IAM::Role",
    "Properties": {
        "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Effect": "Allow",
                    "Principal": {
                        "Service": [
                            "ec2.amazonaws.com"
                        ]
                    },
                    "Action": [
                        "sts:AssumeRole"
                    ]
                }
            ]
        },
        "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

```

AWS::ElasticBeanstalk::ApplicationVersion: '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

```

Fragmentos de plantillas de Elastic Load Balancing

Para crear un Equilibrador de carga de aplicación, un Equilibrador de carga de red o un equilibrador de carga de puerta de enlace, utilice los tipos de recursos de V2, que comienzan con `AWS::ElasticLoadBalancingV2`. Para crear un Equilibrador de carga clásico, use los tipos de recursos que empiezan con `AWS::ElasticLoadBalancing`.

Contenido

- [Recursos de ELBv2](#)
- [Recursos del Equilibrador de carga clásico](#)

Recursos de ELBv2

En este ejemplo se define un Equilibrador de carga de aplicación con un oyente HTTP y una acción predeterminada que reenvía el tráfico al grupo de destino. El equilibrador de carga usa la configuración de comprobación de estado predeterminada. El grupo objetivo tiene dos instancias de EC2 registradas.

YAML

```
Resources:
  myLoadBalancer:
    Type: AWS::ElasticLoadBalancingV2::LoadBalancer
    Properties:
      Name: my-alb
      Type: application
      Scheme: internal
      Subnets:
        - !Ref subnet-AZ1
        - !Ref subnet-AZ2
      SecurityGroups:
        - !Ref mySecurityGroup

  myHTTPListener:
    Type: AWS::ElasticLoadBalancingV2::Listener
    Properties:
      LoadBalancerArn: !Ref myLoadBalancer
      Protocol: HTTP
      Port: 80
      DefaultActions:
        - Type: "forward"
          TargetGroupArn: !Ref myTargetGroup

  myTargetGroup:
    Type: AWS::ElasticLoadBalancingV2::TargetGroup
    Properties:
      Name: "my-target-group"
      Protocol: HTTP
      Port: 80
      TargetType: instance
```

```
VpcId: !Ref myVPC
Targets:
  - Id: !GetAtt Instance1.InstanceId
    Port: 80
  - Id: !GetAtt Instance2.InstanceId
    Port: 80
```

JSON

```
{
  "Resources": {
    "myLoadBalancer": {
      "Type": "AWS::ElasticLoadBalancingV2::LoadBalancer",
      "Properties": {
        "Name": "my-alb",
        "Type": "application",
        "Scheme": "internal",
        "Subnets": [
          {
            "Ref": "subnet-AZ1"
          },
          {
            "Ref": "subnet-AZ2"
          }
        ],
        "SecurityGroups": [
          {
            "Ref": "mySecurityGroup"
          }
        ]
      }
    },
    "myHTTPListener": {
      "Type": "AWS::ElasticLoadBalancingV2::Listener",
      "Properties": {
        "LoadBalancerArn": {
          "Ref": "myLoadBalancer"
        },
        "Protocol": "HTTP",
        "Port": 80,
        "DefaultActions": [
          {
            "Type": "forward",
```

```
        "TargetGroupArn": {
            "Ref": "myTargetGroup"
        }
    },
    ]
},
"myTargetGroup": {
    "Type": "AWS::ElasticLoadBalancingV2::TargetGroup",
    "Properties": {
        "Name": "my-target-group",
        "Protocol": "HTTP",
        "Port": 80,
        "TargetType": "instance",
        "VpcId": {
            "Ref": "myVPC"
        },
        "Targets": [
            {
                "Id": {
                    "Fn::GetAtt": [
                        "Instance1",
                        "InstanceId"
                    ]
                },
                "Port": 80
            },
            {
                "Id": {
                    "Fn::GetAtt": [
                        "Instance2",
                        "InstanceId"
                    ]
                },
                "Port": 80
            }
        ]
    }
}
}
```

Recursos del Equilibrador de carga clásico

En este ejemplo se define un Equilibrador de carga clásico con un oyente HTTP y sin instancias de EC2 registradas. El equilibrador de carga usa la configuración de comprobación de estado predeterminada.

YAML

```
myLoadBalancer:
  Type: AWS::ElasticLoadBalancing::LoadBalancer
  Properties:
    AvailabilityZones:
      - "us-east-1a"
    Listeners:
      - LoadBalancerPort: '80'
        InstancePort: '80'
        Protocol: HTTP
```

JSON

```
"myLoadBalancer" : {
  "Type" : "AWS::ElasticLoadBalancing::LoadBalancer",
  "Properties" : {
    "AvailabilityZones" : [ "us-east-1a" ],
    "Listeners" : [ {
      "LoadBalancerPort" : "80",
      "InstancePort" : "80",
      "Protocol" : "HTTP"
    } ]
  }
}
```

En este ejemplo se define un Equilibrador de carga clásico con un oyente HTTP, dos instancias de EC2 registradas y una configuración de comprobación de estado personalizada.

YAML

```
myClassicLoadBalancer:
  Type: AWS::ElasticLoadBalancing::LoadBalancer
  Properties:
    AvailabilityZones:
```

```
- "us-east-1a"
Instances:
- Ref: Instance1
- Ref: Instance2
Listeners:
- LoadBalancerPort: '80'
  InstancePort: '80'
  Protocol: HTTP
HealthCheck:
  Target: HTTP:80/
  HealthyThreshold: '3'
  UnhealthyThreshold: '5'
  Interval: '30'
  Timeout: '5'
```

JSON

```
"myClassicLoadBalancer" : {
  "Type" : "AWS::ElasticLoadBalancing::LoadBalancer",
  "Properties" : {
    "AvailabilityZones" : [ "us-east-1a" ],
    "Instances" : [
      { "Ref" : "Instance1" },
      { "Ref" : "Instance2" }
    ],
    "Listeners" : [ {
      "LoadBalancerPort" : "80",
      "InstancePort" : "80",
      "Protocol" : "HTTP"
    } ],
    "HealthCheck" : {
      "Target" : "HTTP:80/",
      "HealthyThreshold" : "3",
      "UnhealthyThreshold" : "5",
      "Interval" : "30",
      "Timeout" : "5"
    }
  }
}
```


AWS Identity and Access ManagementFragmentos de plantillas de

Esta sección contiene fragmentos de la plantilla AWS Identity and Access Management.

Temas

- [Declaración de recurso de un usuario de IAM](#)
- [Declaración de recurso de una clave de acceso IAM](#)
- [Declaración de recurso de un grupo de IAM](#)
- [Añadir usuarios a un grupo](#)
- [Declaración de una política de IAM](#)
- [Declaración de una política de bucket de Amazon S3](#)
- [Declaración de una política de tema de Amazon SNS](#)
- [Declaración de una política de Amazon SQS](#)
- [Ejemplos de plantillas de rol de IAM](#)

Important

Al crear o actualizar una pila utilizando una plantilla que contenga recursos de IAM, debe reconocer el uso de las capacidades de IAM. Para obtener más información, consulte [Reconocimiento de recursos de IAM en plantillas de CloudFormation](#).

Declaración de recurso de un usuario de IAM

Este fragmento muestra cómo declarar un recurso de [AWS::IAM::User](#) para crear un usuario de IAM. El usuario está declarado con la ruta ("/") y un perfil de inicio de sesión con la contraseña (myP@ssW0rd).

El documento de políticas denominado giveaccesstoqueueonly ofrece al usuario permiso para realizar todas las acciones de Amazon SQS en el recurso de cola de Amazon SQS myqueue y deniega el acceso a todos los demás recursos de la cola de Amazon SQS. La función Fn::GetAtt obtiene el atributo Arn del recurso [AWS::SQS::Queue](#) myqueue.

El documento de políticas denominado giveaccesstotopiconly se añade al usuario para darle permiso para realizar todas las acciones de Amazon SNS en el recurso de tema de Amazon SNS

mytopic y para denegar el acceso a todos los demás recursos de cola de Amazon SNS. La función `Ref` [AWS::SNS::Topic](#) obtiene el ARN del recurso `mytopic`.

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
```

Declaración de recurso de una clave de acceso IAM

Este fragmento muestra un recurso [AWS::IAM::AccessKey](#). El recurso `myaccesskey` crea una clave de acceso y la asigna a un usuario de IAM que se declare como un recurso [AWS::IAM::User](#) en la plantilla.

JSON

```
"myaccesskey" : {
  "Type" : "AWS::IAM::AccessKey",
  "Properties" : {
    "UserName" : { "Ref" : "myuser" }
  }
}
```

YAML

```
myaccesskey:
  Type: AWS::IAM::AccessKey
  Properties:
    UserName:
      !Ref myuser
```

Puede obtener la clave secreta de un recurso `AWS::IAM::AccessKey` utilizando la función `Fn::GetAtt`. Una forma de recuperar la clave secreta es ponerla en un valor `Output`. Puede obtener la clave de acceso utilizando la función `Ref`. Las siguientes declaraciones de valor `Output` obtiene la clave de acceso y la clave secreta de `myaccesskey`.

JSON

```
"AccessKeyformyaccesskey" : {
  "Value" : { "Ref" : "myaccesskey" }
},
"SecretKeyformyaccesskey" : {
  "Value" : {
    "Fn::GetAtt" : [ "myaccesskey", "SecretAccessKey" ]
  }
}
```

YAML

```
AccessKeyformyaccesskey:
  Value:
    !Ref myaccesskey
SecretKeyformyaccesskey:
  Value: !GetAtt myaccesskey.SecretAccessKey
```

También puede pasar la clave de acceso y la clave secreta de AWS a una instancia de Amazon EC2 o un grupo de escalado automático definidos en la plantilla. La siguiente declaración [AWS::EC2::Instance](#) utiliza la propiedad `UserData` para pasar la clave de acceso y su clave secreta para el recurso `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}"
```

Declaración de recurso de un grupo de IAM

Este fragmento muestra un recurso [AWS::IAM::Group](#). El grupo tiene una ruta ("/myapplication/"). El documento de políticas denominado myapppolicy se añade al grupo para permitir a los usuarios del grupo realizar todas las acciones de Amazon SQS en el recurso de cola de Amazon SQS y para denegar el acceso a todos los demás recursos de Amazon SQS excepto myqueue.

Para asignar una política a un recurso, IAM requiere el nombre de recurso de Amazon (ARN) para el recurso. En el fragmento, la función `Fn::GetAtt` obtiene el ARN de la cola del recurso [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

```

Añadir usuarios a un grupo

El recurso [AWS::IAM::UserToGroupAddition](#) añade usuarios a un grupo. En el siguiente fragmento, el recurso `addUserToGroup` añade los siguientes usuarios a un grupo existente denominado `myexistinggroup2`: el usuario existente `existinguser1` y el usuario `myuser` que se declara como recurso [AWS::IAM::User](#) en la plantilla.

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
```

Declaración de una política de IAM

Este fragmento muestra cómo crear una política y aplicarla a varios grupos con un recurso [AWS::IAM::Policy](#) denominado `mypolicy`. El recurso `mypolicy` contiene una propiedad `PolicyDocument` que permite acciones `GetObject`, `PutObject` y `PutObjectAcl` en los objetos del bucket de S3 representadas por el ARN `arn:aws:s3:::myAWSBucket`. El recurso `mypolicy` aplica la política a un grupo existente denominado `myexistinggroup1` y un grupo `mygroup` declarado en la plantilla como recurso [AWS::IAM::Group](#). Este ejemplo muestra cómo aplicar una política a un grupo con la propiedad `Groups`; sin embargo, también puede usar la propiedad `Users` para añadir un documento de políticas a una lista de usuarios.

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
```

Declaración de una política de bucket de Amazon S3

Este fragmento muestra cómo crear una política y aplicarla a un bucket de Amazon S3 usando el recurso [AWS::S3::BucketPolicy](#). El recurso mybucketpolicy declara un documento de políticas que permite al usuario user1 de IAM realizar la acción GetObject en todos los objetos del bucket de S3 para que se aplique esta política. En el fragmento, la función Fn::GetAtt obtiene el ARN del recurso user1. El recurso mybucketpolicy aplica la política al recurso AWS::S3::BucketPolicy mybucket. La función Ref obtiene el nombre del bucket del recurso 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

```

Declaración de una política de tema de Amazon SNS

Este fragmento muestra cómo crear una política y aplicarla a un tema de Amazon SNS usando el recurso [AWS::SNS::TopicPolicy](#). El recurso `mysnspolicy` contiene una propiedad `PolicyDocument` que permite al recurso [AWS::IAM::User](#) `myuser` realizar la acción `Publish` sobre un recurso [AWS::SNS::Topic](#) `mytopic`. En el fragmento, la función `Fn::GetAtt` obtiene el ARN del recurso `myuser` y la función `Ref` obtiene el ARN para el recurso `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

```

Declaración de una política de Amazon SQS

Este fragmento muestra cómo crear una política y aplicarla a una cola de Amazon SQS usando el recurso [AWS::SQS::QueuePolicy](#). La propiedad `PolicyDocument` permite al usuario existente `myapp` (especificado por su ARN) realizar la acción `SendMessage` en una cola existente (que

especifica su URL) y un recurso [AWS::SQS::Queue](#) myqueue. La función [Ref](#) obtiene el ARN del recurso 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
```

Ejemplos de plantillas de rol de IAM

Esta sección proporciona ejemplos de plantillas de CloudFormation para los roles de IAM de instancias de EC2.

Para obtener más información, consulte [Roles de IAM para Amazon EC2](#) en la Guía del usuario de Amazon EC2.

Rol de IAM con EC2

En este ejemplo, la propiedad `IamInstanceProfile` de la instancia de EC2 hace referencia al perfil de instancia. Tanto la política de instancias como la política de roles hacen referencia a [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
```

Rol de IAM con grupo de escalado automático

En este ejemplo, la propiedad `IamInstanceProfile` de la configuración de inicialización de Amazon EC2 Auto Scaling hace referencia al perfil de instancia.

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 LambdaPlantilla de

La siguiente plantilla utiliza un recurso personalizado y la función de AWS Lambda (Lambda) para adjuntar un nuevo grupo de seguridad a una lista de grupos de seguridad existentes. Esta función es útil cuando se quiere crear una lista de los grupos de seguridad de forma dinámica, de forma

que la lista incluye tanto los grupos de seguridad nuevos como ya existentes. Por ejemplo, puede transferir una lista de los grupos de seguridad existentes como parámetro de valor, añadir el nuevo valor a la lista y, a continuación, asociar todos los valores a una instancia de EC2. Para obtener más información acerca del tipo de recurso de una función de Lambda, consulte [AWS::Lambda::Function](#).

En el ejemplo, cuando CloudFormation crea el recurso personalizado AllSecurityGroups, CloudFormation invoca la función de Lambda AppendItemToListFunction. CloudFormation pasa la lista de los grupos de seguridad existentes y un nuevo grupo de seguridad (NewSecurityGroup) a la función, que adjunta el nuevo grupo de seguridad a la lista y, a continuación, devuelve la lista modificada. CloudFormation utiliza la lista modificada para asociar todos los grupos de seguridad al recurso de MyEC2Instance.

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Parameters": {
    "ExistingSecurityGroups": {
      "Type": "List<AWS::EC2::SecurityGroup::Id>"
    },
    "ExistingVPC": {
      "Type": "AWS::EC2::VPC::Id",
      "Description": "The VPC ID that includes the security groups in the ExistingSecurityGroups parameter."
    },
    "InstanceType": {
      "Type": "String",
      "Default": "t2.micro",
      "AllowedValues": [
        "t2.micro",
        "t3.micro"
      ]
    }
  },
  "Resources": {
    "SecurityGroup": {
      "Type": "AWS::EC2::SecurityGroup",
      "Properties": {
        "GroupDescription": "Allow HTTP traffic to the host",
        "VpcId": {
          "Ref": "ExistingVPC"
        }
      }
    }
  }
}
```

```

    },
    "SecurityGroupIngress": [
      {
        "IpProtocol": "tcp",
        "FromPort": 80,
        "ToPort": 80,
        "CidrIp": "0.0.0.0/0"
      }
    ],
    "SecurityGroupEgress": [
      {
        "IpProtocol": "tcp",
        "FromPort": 80,
        "ToPort": 80,
        "CidrIp": "0.0.0.0/0"
      }
    ]
  },
  "AllSecurityGroups": {
    "Type": "Custom::Split",
    "Properties": {
      "ServiceToken": {
        "Fn::GetAtt": [
          "AppendItemToListFunction",
          "Arn"
        ]
      },
      "List": {
        "Ref": "ExistingSecurityGroups"
      },
      "AppendedItem": {
        "Ref": "SecurityGroup"
      }
    }
  },
  "AppendItemToListFunction": {
    "Type": "AWS::Lambda::Function",
    "Properties": {
      "Handler": "index.handler",
      "Role": {
        "Fn::GetAtt": [
          "LambdaExecutionRole",
          "Arn"
        ]
      }
    }
  }
}

```

```

        ]
      },
      "Code": {
        "ZipFile": {
          "Fn::Join": [
            "",
            [
              "var response = require('cfn-response');",
              "exports.handler = function(event, context) {",
              "  var responseData = {Value:",
event.ResourceProperties.List};",
              "  ",
              responseData.Value.push(event.ResourceProperties.AppendedItem);",
              "  response.send(event, context, response.SUCCESS,",
              responseData);",
              "};"
            ]
          ]
        },
        "Runtime": "nodejs20.x"
      }
    },
    "MyEC2Instance": {
      "Type": "AWS::EC2::Instance",
      "Properties": {
        "ImageId": "{{resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2}}",
        "SecurityGroupIds": {
          "Fn::GetAtt": [
            "AllSecurityGroups",
            "Value"
          ]
        },
        "InstanceType": {
          "Ref": "InstanceType"
        }
      }
    },
    "LambdaExecutionRole": {
      "Type": "AWS::IAM::Role",
      "Properties": {
        "AssumeRolePolicyDocument": {
          "Version": "2012-10-17",

```

```

        "Statement": [
            {
                "Effect": "Allow",
                "Principal": {
                    "Service": [
                        "lambda.amazonaws.com"
                    ]
                },
                "Action": [
                    "sts:AssumeRole"
                ]
            }
        ],
        "Path": "/",
        "Policies": [
            {
                "PolicyName": "root",
                "PolicyDocument": {
                    "Version": "2012-10-17",
                    "Statement": [
                        {
                            "Effect": "Allow",
                            "Action": [
                                "logs:*"
                            ],
                            "Resource": "arn:aws:logs:*:*:*"
                        }
                    ]
                }
            }
        ]
    },
    "Outputs": {
        "AllSecurityGroups": {
            "Description": "Security Groups that are associated with the EC2 instance",
            "Value": {
                "Fn::Join": [
                    ", ",
                    {
                        "Fn::GetAtt": [
                            "AllSecurityGroups",

```

```
}  
}  
}  
}  
]  
}  
]  
]"Value"
```

YAML

```

AWSTemplateFormatVersion: '2010-09-09'

Parameters:
  ExistingSecurityGroups:
    Type: List<AWS::EC2::SecurityGroup::Id>
  ExistingVPC:
    Type: AWS::EC2::VPC::Id
    Description: The VPC ID that includes the security groups in the
ExistingSecurityGroups parameter.
  InstanceType:
    Type: String
    Default: t2.micro
    AllowedValues:
      - t2.micro
      - t3.micro

Resources:
  SecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupDescription: Allow HTTP traffic to the host
      VpcId: !Ref ExistingVPC
      SecurityGroupIngress:
        - IpProtocol: tcp
          FromPort: 80
          ToPort: 80
          CidrIp: 0.0.0.0/0
      SecurityGroupEgress:
        - IpProtocol: tcp
          FromPort: 80
          ToPort: 80
          CidrIp: 0.0.0.0/0
  AllSecurityGroups:
    Type: Custom::Split

```

```

Properties:
  ServiceToken: !GetAtt AppendItemToListFunction.Arn
  List: !Ref ExistingSecurityGroups
  AppendedItem: !Ref SecurityGroup
AppendItemToListFunction:
  Type: AWS::Lambda::Function
  Properties:
    Handler: index.handler
    Role: !GetAtt LambdaExecutionRole.Arn
    Code:
      ZipFile: !Join
        - ''
        - - var response = require('cfn-response');
          - exports.handler = function(event, context) {
            - '   var responseData = {Value: event.ResourceProperties.List};'
            - '   responseData.Value.push(event.ResourceProperties.AppendedItem);'
            - '   response.send(event, context, response.SUCCESS, responseData);'
            - '};'
    Runtime: nodejs20.x
MyEC2Instance:
  Type: AWS::EC2::Instance
  Properties:
    ImageId: '{{resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-
x86_64-gp2}}'
    SecurityGroupIds: !GetAtt AllSecurityGroups.Value
    InstanceType: !Ref InstanceType
LambdaExecutionRole:
  Type: AWS::IAM::Role
  Properties:
    AssumeRolePolicyDocument:
      Version: '2012-10-17'
      Statement:
        - Effect: Allow
          Principal:
            Service:
              - lambda.amazonaws.com
          Action:
            - sts:AssumeRole
    Path: /
    Policies:
      - PolicyName: root
        PolicyDocument:
          Version: '2012-10-17'
          Statement:

```



```
- Effect: Allow
  Action:
    - logs:*
  Resource: arn:aws:logs:*:*:*

Outputs:
  AllSecurityGroups:
    Description: Security Groups that are associated with the EC2 instance
    Value: !Join
      - ', '
      - !GetAtt AllSecurityGroups.Value
```

Fragmentos de plantillas de Amazon Redshift

Amazon Redshift es un servicio de almacenamiento de datos administrado de varios petabytes en la nube. Puede usar CloudFormation para aprovisionar y administrar clústeres de Amazon Redshift.

Clúster de Amazon Redshift

La siguiente plantilla de ejemplo crea un clúster de Amazon Redshift de acuerdo con los valores de los parámetros que se especifican cuando se crea la pila. El grupo de parámetros del clúster que se asocia con el clúster de Amazon Redshift permite el registro de la actividad del usuario. La plantilla también lanza clústeres de Amazon Redshift en una Amazon VPC que se define en la plantilla. La VPC incluye una gateway de Internet para que pueda acceder a los clústeres de Amazon Redshift desde Internet. Sin embargo, la comunicación entre el clúster y el puerto de enlace a Internet deben estar habilitada, lo que se realiza en la tabla de ruteo.

Note

La plantilla incluye la condición `IsMultiNodeCluster`, de modo que el parámetro `NumberOfNodes` se declara solo cuando el valor del parámetro `ClusterType` está establecido en `multi-node`.

El ejemplo define el parámetro `MySQLRootPassword` con la propiedad `NoEcho` establecida en `true`. Si establece el atributo `NoEcho` en `true`, CloudFormation devuelve el valor del parámetro enmascarado como asteriscos (`*****`) para cualquier llamada que describa la pila o los eventos de la pila, excepto para la información almacenada en las ubicaciones especificadas a continuación.

⚠ Important

El uso del atributo NoEcho no enmascara ninguna información almacenada en lo que se muestra a continuación:

- La sección de la plantilla Metadata. CloudFormation no transforma, modifica ni redacta ninguna información que incluya en la sección Metadata. Para obtener más información, consulte [Metadata](#).
- La sección de la plantilla Outputs. Para obtener más información, consulte [Salidas](#).
- El atributo Metadata de una definición de recurso. Para obtener más información, consulte [Metadata atributo](#).

Recomendamos encarecidamente que no utilice estos mecanismos para incluir información confidencial, como contraseñas o secretos.

⚠ Important

En lugar de integrar información confidencial directamente en las plantillas de CloudFormation, se recomienda utilizar parámetros dinámicos en la plantilla de la pila para hacer referencia a la información confidencial almacenada y administrada fuera de CloudFormation, como en AWS Systems Manager Parameter Store o AWS Secrets Manager.

Para obtener más información, consulte la práctica recomendada [No integre credenciales en sus plantillas](#).

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Parameters" : {
    "DatabaseName" : {
      "Description" : "The name of the first database to be created when the cluster is
created",
      "Type" : "String",
      "Default" : "dev",
      "AllowedPattern" : "([a-z]|[0-9])+"
    }
  }
}
```

```

},
"ClusterType" : {
  "Description" : "The type of cluster",
  "Type" : "String",
  "Default" : "single-node",
  "AllowedValues" : [ "single-node", "multi-node" ]
},
"NumberOfNodes" : {
  "Description" : "The number of compute nodes in the cluster. For multi-node
clusters, the NumberOfNodes parameter must be greater than 1",
  "Type" : "Number",
  "Default" : "1"
},
"NodeType" : {
  "Description" : "The type of node to be provisioned",
  "Type" : "String",
  "Default" : "ds2.xlarge",
  "AllowedValues" : [ "ds2.xlarge", "ds2.8xlarge", "dc1.large", "dc1.8xlarge" ]
},
"MasterUsername" : {
  "Description" : "The user name that is associated with the master user account
for the cluster that is being created",
  "Type" : "String",
  "Default" : "defaultuser",
  "AllowedPattern" : "([a-z])([a-z]|[0-9])*"
},
"MasterUserPassword" : {
  "Description" : "The password that is associated with the master user account for
the cluster that is being created.",
  "Type" : "String",
  "NoEcho" : "true"
},
"InboundTraffic" : {
  "Description" : "Allow inbound traffic to the cluster from this CIDR range.",
  "Type" : "String",
  "MinLength" : "9",
  "MaxLength" : "18",
  "Default" : "0.0.0.0/0",
  "AllowedPattern" : "(\\d{1,3})\\. (\\d{1,3})\\. (\\d{1,3})\\. (\\d{1,3})/ (\\d{1,2})",
  "ConstraintDescription" : "must be a valid CIDR range of the form x.x.x.x/x."
},
"PortNumber" : {

```

```

    "Description" : "The port number on which the cluster accepts incoming
connections.",
    "Type" : "Number",
    "Default" : "5439"
  }
},
"Conditions" : {
  "IsMultiNodeCluster" : {
    "Fn::Equals" : [{ "Ref" : "ClusterType" }, "multi-node" ]
  }
},
"Resources" : {
  "RedshiftCluster" : {
    "Type" : "AWS::Redshift::Cluster",
    "DependsOn" : "AttachGateway",
    "Properties" : {
      "ClusterType" : { "Ref" : "ClusterType" },
      "NumberOfNodes" : { "Fn::If" : [ "IsMultiNodeCluster", { "Ref" :
"NumberOfNodes" }, { "Ref" : "AWS::NoValue" } ] },
      "NodeType" : { "Ref" : "NodeType" },
      "DBName" : { "Ref" : "DatabaseName" },
      "MasterUsername" : { "Ref" : "MasterUsername" },
      "MasterUserPassword" : { "Ref" : "MasterUserPassword" },
      "ClusterParameterGroupName" : { "Ref" : "RedshiftClusterParameterGroup" },
      "VpcSecurityGroupIds" : [ { "Ref" : "SecurityGroup" } ],
      "ClusterSubnetGroupName" : { "Ref" : "RedshiftClusterSubnetGroup" },
      "PubliclyAccessible" : "true",
      "Port" : { "Ref" : "PortNumber" }
    }
  },
  "RedshiftClusterParameterGroup" : {
    "Type" : "AWS::Redshift::ClusterParameterGroup",
    "Properties" : {
      "Description" : "Cluster parameter group",
      "ParameterGroupFamily" : "redshift-1.0",
      "Parameters" : [{
        "ParameterName" : "enable_user_activity_logging",
        "ParameterValue" : "true"
      }]
    }
  },
  "RedshiftClusterSubnetGroup" : {
    "Type" : "AWS::Redshift::ClusterSubnetGroup",
    "Properties" : {

```

```
    "Description" : "Cluster subnet group",
    "SubnetIds" : [ { "Ref" : "PublicSubnet" } ]
  },
  "VPC" : {
    "Type" : "AWS::EC2::VPC",
    "Properties" : {
      "CidrBlock" : "10.0.0.0/16"
    }
  },
  "PublicSubnet" : {
    "Type" : "AWS::EC2::Subnet",
    "Properties" : {
      "CidrBlock" : "10.0.0.0/24",
      "VpcId" : { "Ref" : "VPC" }
    }
  },
  "SecurityGroup" : {
    "Type" : "AWS::EC2::SecurityGroup",
    "Properties" : {
      "GroupDescription" : "Security group",
      "SecurityGroupIngress" : [ {
        "CidrIp" : { "Ref" : "InboundTraffic" },
        "FromPort" : { "Ref" : "PortNumber" },
        "ToPort" : { "Ref" : "PortNumber" },
        "IpProtocol" : "tcp"
      } ],
      "VpcId" : { "Ref" : "VPC" }
    }
  },
  "myInternetGateway" : {
    "Type" : "AWS::EC2::InternetGateway"
  },
  "AttachGateway" : {
    "Type" : "AWS::EC2::VPCGatewayAttachment",
    "Properties" : {
      "VpcId" : { "Ref" : "VPC" },
      "InternetGatewayId" : { "Ref" : "myInternetGateway" }
    }
  },
  "PublicRouteTable" : {
    "Type" : "AWS::EC2::RouteTable",
    "Properties" : {
      "VpcId" : {
```

```

        "Ref" : "VPC"
    }
}
},
"PublicRoute" : {
    "Type" : "AWS::EC2::Route",
    "DependsOn" : "AttachGateway",
    "Properties" : {
        "RouteTableId" : {
            "Ref" : "PublicRouteTable"
        },
        "DestinationCidrBlock" : "0.0.0.0/0",
        "GatewayId" : {
            "Ref" : "myInternetGateway"
        }
    }
},
"PublicSubnetRouteTableAssociation" : {
    "Type" : "AWS::EC2::SubnetRouteTableAssociation",
    "Properties" : {
        "SubnetId" : {
            "Ref" : "PublicSubnet"
        },
        "RouteTableId" : {
            "Ref" : "PublicRouteTable"
        }
    }
},
"Outputs" : {
    "ClusterEndpoint" : {
        "Description" : "Cluster endpoint",
        "Value" : { "Fn::Join" : [ ":", [ { "Fn::GetAtt" : [ "RedshiftCluster",
"Endpoint.Address" ] }, { "Fn::GetAtt" : [ "RedshiftCluster",
"Endpoint.Port" ] } ] ] }
    },
    "ClusterName" : {
        "Description" : "Name of cluster",
        "Value" : { "Ref" : "RedshiftCluster" }
    },
    "ParameterGroupName" : {
        "Description" : "Name of parameter group",
        "Value" : { "Ref" : "RedshiftClusterParameterGroup" }
    }
},

```

```

    "RedshiftClusterSubnetGroupName" : {
      "Description" : "Name of cluster subnet group",
      "Value" : { "Ref" : "RedshiftClusterSubnetGroup" }
    },
    "RedshiftClusterSecurityGroupName" : {
      "Description" : "Name of cluster security group",
      "Value" : { "Ref" : "SecurityGroup" }
    }
  }
}

```

YAML

```

AWSTemplateFormatVersion: '2010-09-09'
Parameters:
  DatabaseName:
    Description: The name of the first database to be created when the cluster is
      created
    Type: String
    Default: dev
    AllowedPattern: "([a-z]|[0-9])+\"
  ClusterType:
    Description: The type of cluster
    Type: String
    Default: single-node
    AllowedValues:
      - single-node
      - multi-node
  NumberOfNodes:
    Description: The number of compute nodes in the cluster. For multi-node clusters,
      the NumberOfNodes parameter must be greater than 1
    Type: Number
    Default: '1'
  NodeType:
    Description: The type of node to be provisioned
    Type: String
    Default: ds2.xlarge
    AllowedValues:
      - ds2.xlarge
      - ds2.8xlarge
      - dc1.large
      - dc1.8xlarge
  MasterUsername:

```

Description: The user name that is associated with the master user account for the cluster that is being created

Type: String

Default: defaultuser

AllowedPattern: "([a-z])([a-z]|[0-9])*"

MasterUserPassword:

Description: The password that is associated with the master user account for the cluster that is being created.

Type: String

NoEcho: 'true'

InboundTraffic:

Description: Allow inbound traffic to the cluster from this CIDR range.

Type: String

MinLength: '9'

MaxLength: '18'

Default: 0.0.0.0/0

AllowedPattern: "(\\d{1,3})\\. (\\d{1,3})\\. (\\d{1,3})\\. (\\d{1,3})/(\\d{1,2})"

ConstraintDescription: must be a valid CIDR range of the form x.x.x.x/x.

PortNumber:

Description: The port number on which the cluster accepts incoming connections.

Type: Number

Default: '5439'

Conditions:

IsMultiNodeCluster:

Fn::Equals:

- Ref: ClusterType
- multi-node

Resources:

RedshiftCluster:

Type: AWS::Redshift::Cluster

DependsOn: AttachGateway

Properties:

ClusterType:

Ref: ClusterType

NumberOfNodes:

Fn::If:

- IsMultiNodeCluster
- Ref: NumberOfNodes
- Ref: AWS::NoValue

NodeType:

Ref: NodeType

DBName:

Ref: DatabaseName

MasterUsername:


```
    Ref: MasterUsername
MasterUserPassword:
    Ref: MasterUserPassword
ClusterParameterGroupName:
    Ref: RedshiftClusterParameterGroup
VpcSecurityGroupIds:
    - Ref: SecurityGroup
ClusterSubnetGroupName:
    Ref: RedshiftClusterSubnetGroup
PubliclyAccessible: 'true'
Port:
    Ref: PortNumber
RedshiftClusterParameterGroup:
    Type: AWS::Redshift::ClusterParameterGroup
    Properties:
        Description: Cluster parameter group
        ParameterGroupFamily: redshift-1.0
        Parameters:
            - ParameterName: enable_user_activity_logging
              ParameterValue: 'true'
RedshiftClusterSubnetGroup:
    Type: AWS::Redshift::ClusterSubnetGroup
    Properties:
        Description: Cluster subnet group
        SubnetIds:
            - Ref: PublicSubnet
VPC:
    Type: AWS::EC2::VPC
    Properties:
        CidrBlock: 10.0.0.0/16
PublicSubnet:
    Type: AWS::EC2::Subnet
    Properties:
        CidrBlock: 10.0.0.0/24
        VpcId:
            Ref: VPC
SecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
        GroupDescription: Security group
        SecurityGroupIngress:
            - CidrIp:
                Ref: InboundTraffic
              FromPort:
```

```
        Ref: PortNumber
    ToPort:
        Ref: PortNumber
    IpProtocol: tcp
    VpcId:
        Ref: VPC
myInternetGateway:
    Type: AWS::EC2::InternetGateway
AttachGateway:
    Type: AWS::EC2::VPCGatewayAttachment
    Properties:
        VpcId:
            Ref: VPC
        InternetGatewayId:
            Ref: myInternetGateway
PublicRouteTable:
    Type: AWS::EC2::RouteTable
    Properties:
        VpcId:
            Ref: VPC
PublicRoute:
    Type: AWS::EC2::Route
    DependsOn: AttachGateway
    Properties:
        RouteTableId:
            Ref: PublicRouteTable
        DestinationCidrBlock: 0.0.0.0/0
        GatewayId:
            Ref: myInternetGateway
PublicSubnetRouteTableAssociation:
    Type: AWS::EC2::SubnetRouteTableAssociation
    Properties:
        SubnetId:
            Ref: PublicSubnet
        RouteTableId:
            Ref: PublicRouteTable
Outputs:
    ClusterEndpoint:
        Description: Cluster endpoint
        Value: !Sub "${RedshiftCluster.Endpoint.Address}:${RedshiftCluster.Endpoint.Port}"
    ClusterName:
        Description: Name of cluster
        Value:
            Ref: RedshiftCluster
```

```
ParameterGroupName:
  Description: Name of parameter group
  Value:
    Ref: RedshiftClusterParameterGroup
RedshiftClusterSubnetGroupName:
  Description: Name of cluster subnet group
  Value:
    Ref: RedshiftClusterSubnetGroup
RedshiftClusterSecurityGroupName:
  Description: Name of cluster security group
  Value:
    Ref: SecurityGroup
```

Véase también

[AWS::Redshift::Cluster](#)

Fragmentos de las plantillas de Amazon RDS

Temas

- [Recurso de instancia de la base de datos de Amazon RDS](#)
- [Recurso de instancia de base de datos de Oracle de Amazon RDS](#)
- [Recurso de DBSecurityGroup de Amazon RDS para rango de CIDR](#)
- [DBSecurityGroup de Amazon RDS con un grupo de seguridad de Amazon EC2](#)
- [Múltiples grupos de seguridad de VPC](#)
- [Instancia de base de datos de Amazon RDS en un grupo de seguridad de VPC](#)

Recurso de instancia de la base de datos de Amazon RDS

En este ejemplo se muestra un recurso de instancia de base de datos de Amazon RDS con contraseña de usuario maestra administrada. Para obtener más información, consulte [Administración de contraseñas con AWS Secrets Manager](#) en la Guía del usuario de Amazon RDS y [Administración de contraseñas con AWS Secrets Manager](#) en la Guía del usuario de Aurora. Dado que no se especifica la propiedad `EngineVersion` opcional, se usa la versión del motor predeterminado para esta instancia de base de datos. Para más detalles acerca de la versión del motor predeterminado y otros ajustes predeterminados, consulte [CreateDBInstance](#). La propiedad `DBSecurityGroups` autoriza la entrada de la red a los recursos `AWS::RDS::DBSecurityGroup` denominados

MyDbSecurityByEC2SecurityGroup y myDBSecurityByCIDRIPGroup. Para obtener más información, consulte [AWS::RDS::DBInstance](#). El recurso de la instancia de base de datos también tiene un atributo DeletionPolicy y establecido como Snapshot. Con Snapshot establecido en DeletionPolicy, CloudFormation toma una instantánea de esta instancia de base de datos antes de eliminarla durante la eliminación de la pila.

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
```

Recurso de instancia de base de datos de Oracle de Amazon RDS

En este ejemplo se crea un recurso de instancia de base de datos de Oracle Database con contraseña de usuario maestro administrada. Para obtener más información, consulte [Administración de contraseñas con AWS Secrets Manager](#) en la Guía del usuario de Amazon RDS. En el ejemplo se especifica el Engine como `oracle-ee` con un modelo de licencia de tipo traiga su propia licencia. Para más detalles acerca de las instancias de base de datos de Oracle Database, consulte [CreateDBInstance](#). La propiedad `DBSecurityGroups` autoriza el acceso a la red de los recursos de `AWS::RDS::DBSecurityGroup` denominados `MyDbSecurityByEC2SecurityGroup` y `MyDbSecurityByCIDRIPGroup`. Para obtener más información, consulte [AWS::RDS::DBInstance](#). El recurso de la instancia de base de datos también tiene un atributo `DeletionPolicy` y establecido como `Snapshot`. Con `Snapshot` establecido en `DeletionPolicy`, CloudFormation toma una instantánea de esta instancia de base de datos antes de eliminarla durante la eliminación de la pila.

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

```

Recurso de DBSecurityGroup de Amazon RDS para rango de CIDR

En este ejemplo se muestra un recurso DBSecurityGroup de Amazon RDS con autorización de entrada para el rango de CIDR especificado en el formato ddd.ddd.ddd.ddd/dd. Para obtener más información, consulte [AWS::RDS::DBSecurityGroup](#) e [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"

```

DBSecurityGroup de Amazon RDS con un grupo de seguridad de Amazon EC2

En este ejemplo se muestra un recurso [AWS::RDS::DBSecurityGroup](#) con autorización de entrada de un grupo de seguridad de Amazon EC2 al que hace referencia MyEc2SecurityGroup.

Para ello, debe definir un grupo de seguridad de EC2 y, a continuación, usar la función Ref intrínseca para consultar el grupo de seguridad de EC2 en su 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

Este ejemplo está sacado del siguiente ejemplo completo:

[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

```

Múltiples grupos de seguridad de VPC

En este ejemplo se muestra un recurso [AWS::RDS::DBSecurityGroup](#) con autorización de entrada de múltiples grupos de seguridad VPC de Amazon EC2 en [AWS::RDS::DBSecurityGroupIngress](#).

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'
```

Instancia de base de datos de Amazon RDS en un grupo de seguridad de VPC

En este ejemplo se muestra una instancia de base de datos de Amazon RDS asociada a un grupo de seguridad de VPC de Amazon EC2.

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
```

Fragmentos de plantilla de Route 53

Temas

- [Conjunto de registros de recursos de Amazon Route 53 con el ID o nombre de zona hospedada](#)
- [Uso de RecordSetGroup para la configuración de conjuntos de registros de recursos ponderados](#)
- [Uso de RecordSetGroup para configurar un conjunto de registros de recursos de alias](#)
- [Conjunto de registros de recursos de alias para una distribución de CloudFront](#)

Conjunto de registros de recursos de Amazon Route 53 con el ID o nombre de zona hospedada

Al crear un conjunto de registros de recursos de Amazon Route 53, debe especificar la zona alojada donde desee agregarlo. CloudFormation ofrece dos formas de especificar una zona alojada:

- Puede especificar explícitamente la zona hospedada mediante la propiedad `HostedZoneId`.
- Puede solicitar que CloudFormation busque la zona alojada con la propiedad `HostedZoneName`. Si utiliza la propiedad `HostedZoneName` y hay varias zonas alojadas con el mismo nombre, CloudFormation no crea la pila.

Adición de RecordSet mediante `HostedZoneId`

En este ejemplo se agrega un conjunto de registros de recursos de Amazon Route 53 que contiene un registro SPF para el nombre de dominio `mysite.example.com` que utiliza la propiedad `HostedZoneId` para especificar la zona alojada.

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"'
```

Adición de RecordSet mediante HostedZoneName

En este ejemplo se agrega un conjunto de registros de recursos de Amazon Route 53 para el nombre de dominio “mysite.example.com” mediante la propiedad HostedZoneName para especificar la zona hospedada.

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

```

Uso de RecordSetGroup para la configuración de conjuntos de registros de recursos ponderados

En este ejemplo, se utiliza el recurso [AWS::Route53::RecordSetGroup](#) para configurar dos registros CNAME para “example.com”. zona alojada. La propiedad RecordSets contiene los conjuntos de registros de CNAME para el nombre de DNS “mysite.example.com”. Cada conjunto de registros contiene un identificador (SetIdentifier) y un peso (Weight). La proporción de tráfico de Internet que se enruta a los recursos se basa en los siguientes cálculos:

- Frontend One: $140 / (140 + 60) = 140 / 200 = 70 \%$
- Frontend Two: $60 / (140 + 60) = 60 / 200 = 30 \%$

Para obtener más información sobre los conjuntos de registros de recursos ponderados, consulte [Ruteo ponderado](#) en la Guía para desarrolladores de 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

```

Uso de RecordSetGroup para configurar un conjunto de registros de recursos de alias

En los ejemplos siguientes se utiliza [AWS::Route53::RecordSetGroup](#) para configurar un conjunto de registros de recursos de alias denominado `example.com` que enruta el tráfico a un balanceador de carga de ELB, versión 1 (Classic) y a un balanceador de carga de la versión 2 (Application o Network). La propiedad [AliasTarget](#) especifica el ID de zona alojada y el nombre de DNS para myELB LoadBalancer mediante la función intrínseca `GetAtt`. `GetAtt` recupera diferentes propiedades del recurso myELB, en función de si se dirige el tráfico a un equilibrador de carga versión 1 o versión 2:

- Balanceador de carga versión 1: `CanonicalHostedZoneNameID` y `DNSName`
- Balanceador de carga de la versión 2: `CanonicalHostedZoneID` y `DNSName`

Para obtener más información sobre los conjuntos de registros de recursos de alias, consulte [Elección entre registros de alias y sin alias](#) en la Guía para desarrolladores de Route 53.

JSON para balanceador de carga, versión 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 para balanceador de carga, versión 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 para balanceador de carga, versión 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 para balanceador de carga, versión 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'

```

Conjunto de registros de recursos de alias para una distribución de CloudFront

En el siguiente ejemplo se crea un registro de alias A que dirige un nombre de dominio personalizado a una distribución de CloudFront existente. Se asume que `myHostedZoneID` es una referencia a un recurso `AWS::Route53::HostedZone` real en la misma

plantilla o un parámetro. `myCloudFrontDistribution` hace referencia a un recurso `AWS::CloudFront::Distribution` dentro de la misma plantilla. El registro de alias utiliza el ID de zona alojada estándar de CloudFront (`Z2FDTNDATAQYW2`) y resuelve automáticamente el nombre de dominio de la distribución mediante `Fn::GetAtt`. Esta configuración permite enrutar el tráfico web desde el dominio personalizado a la distribución de CloudFront sin necesidad de una dirección IP.

Note

Cuando se crean conjuntos de registros de recursos de alias, debe especificar `Z2FDTNDATAQYW2` para la propiedad `HostedZoneId`. Los conjuntos de registros de recursos de alias para CloudFront no se pueden crear en una zona privada.

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
```

Fragmentos de código de plantillas de Amazon S3

Utilice estas plantillas de ejemplo de Amazon S3 para ayudar a describir los buckets de Amazon S3 con CloudFormation. Para obtener más ejemplos, consulte la sección [Ejemplos](#) del recurso de `AWS::S3::Bucket`.

Temas

- [Creación de un bucket de Amazon S3 con valores predeterminados](#)
- [Creación de un bucket de Amazon S3 para el alojamiento de sitios web y con DeletionPolicy](#)
- [Creación de un sitio web estático con un dominio personalizado](#)

Creación de un bucket de Amazon S3 con valores predeterminados

Este ejemplo utiliza un [AWS::S3::Bucket](#) para crear un bucket con configuración predeterminada.

JSON

```
"myS3Bucket" : {  
  "Type" : "AWS::S3::Bucket"  
}
```

YAML

```
MyS3Bucket:
  Type: AWS::S3::Bucket
```

Creación de un bucket de Amazon S3 para el alojamiento de sitios web y con **DeletionPolicy**

En este ejemplo se crea un bucket como sitio web y se deshabilita el bloqueo de acceso público (los permisos de lectura públicos son necesarios para los buckets configurados para el alojamiento de sitios web). A continuación, se agrega una política de bucket pública al bucket. Dado que el recurso de este bucket tiene un atributo `DeletionPolicy` establecido en `Retain`, CloudFormation no eliminará este bucket cuando elimine la pila. La sección `Output` usa `Fn::GetAtt` para recuperar el atributo `WebsiteURL` y el atributo `DomainName` del recurso `S3Bucket`.

Note

En los ejemplos siguientes se supone que las configuraciones de bloqueo de acceso público `BlockPublicPolicy` y `RestrictPublicBuckets` están deshabilitadas en la cuenta.

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

```

Creación de un sitio web estático con un dominio personalizado

Puede utilizar Route 53 con un dominio registrado. El siguiente ejemplo presupone que ya ha creado una zona alojada en Route 53 para su dominio. El ejemplo crea dos buckets para el alojamiento de sitios web. El bucket raíz hospeda el contenido y los buckets restantes redireccionan las solicitudes de `www.domainname.com` al bucket raíz. Los conjuntos de registros asignan su nombre de dominio a los puntos de enlace de Amazon S3.

También deberá agregar una política de bucket, tal y como se muestra en los ejemplos anteriores.

Para obtener más información acerca del uso de un dominio personalizado, consulte [Tutorial: Configuración de un sitio web estático mediante un dominio personalizado registrado con Route 53](#) en la Guía del usuario de Amazon Simple Storage Service.

Note

En los ejemplos siguientes se supone que las configuraciones de bloqueo de acceso público `BlockPublicPolicy` y `RestrictPublicBuckets` están deshabilitadas en la cuenta.

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: 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

```

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

```

Fragmentos de plantillas de Amazon SNS

Este ejemplo muestra un recurso de tema Amazon SNS. Requiere una dirección de correo electrónico válida.

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

```

Fragmentos de código de plantillas de Amazon SQS

Este ejemplo muestra una cola de Amazon SQS.

JSON

```
"MyQueue" : {  
  "Type" : "AWS::SQS::Queue",  
  "Properties" : {  
    "VisibilityTimeout" : "value"  
  }  
}
```

YAML

```
MyQueue:  
  Type: AWS::SQS::Queue  
  Properties:  
    VisibilityTimeout: value
```

Fragmentos de plantilla de Amazon Timestream

Amazon Timestream para InfluxDB le facilita a los desarrolladores de aplicaciones y a los equipos de DevOps la ejecución de bases de datos de InfluxDB totalmente administradas en AWS para aplicaciones de series temporales en tiempo real mediante las API de código abierto. Puede crear rápidamente una base de datos de InfluxDB que administre cargas de trabajo exigentes basadas en series temporales. Con unas simples llamadas a la API, puede configurar, migrar, operar y escalar una base de datos de InfluxDB en AWS mediante la aplicación automática de parches, copias de seguridad y recuperación de software. También puede encontrar estos ejemplos en [awslabs/amazon-timestream-tools/tree/mainline/integrations/cloudformation/timestream-influxdb](https://aws-labs.github.io/amazon-timestream-tools/tree/mainline/integrations/cloudformation/timestream-influxdb) en GitHub.

Temas

- [Muestra mínima con valores predeterminados](#)
- [Un ejemplo más completo con parámetros](#)

Estas plantillas de CloudFormation crean los siguientes recursos necesarios para crear, conectar y supervisar correctamente una instancia de Amazon Timestream para InfluxDB:

Amazon VPC

- VPC
- Una Subnet o más

- InternetGateway
- RouteTable
- SecurityGroup

Amazon S3

- Bucket

Amazon Timestream

- InfluxDBInstance

Muestra mínima con valores predeterminados

En este ejemplo, se implementa una instancia con múltiples zonas de disponibilidad y de acceso público utilizando los valores predeterminados siempre que sea posible.

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::VPCElasticNetworkAttachment

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"]]

Un ejemplo más completo con parámetros

Esta plantilla de ejemplo altera dinámicamente los recursos de la red en función de los parámetros proporcionados. Los parámetros incluyen a `PubliclyAccessible` y `DeploymentType`.

JSON

```
{
  "Metadata": {
    "AWS::CloudFormation::Interface": {
      "ParameterGroups": [
        {
          "Label": {"default": "Network Configuration"},
          "Parameters": ["VPCCIDR"]
        },
        {
          "Label": {"default": "Amazon Timestream for InfluxDB Configuration"},
          "Parameters": [
            "DbInstanceName",
            "InfluxDBUsername",
            "InfluxDBPassword",
            "InfluxDBOrganization",
            "InfluxDBBucket",
            "DbInstanceType",
            "DbStorageType",
            "AllocatedStorage",
            "PubliclyAccessible",
            "DeploymentType"
          ]
        }
      ]
    },
    "ParameterLabels": {
      "VPCCIDR": {"default": "VPC CIDR"}
    }
  }
}
```

```

    }
  }
},
"Parameters": {
  "VPCCIDR": {
    "Description": "Please enter the IP range (CIDR notation) for the new VPC",
    "Type": "String",
    "Default": "10.0.0.0/16"
  },
  "DbInstanceName": {
    "Description": "The name that uniquely identifies the DB instance when
interacting with the Amazon Timestream for InfluxDB API and CLI commands. This name
will also be a prefix included in the endpoint. DB instance names must be unique per
customer and per Region.",
    "Type": "String",
    "Default": "mydbinstance",
    "MinLength": 3,
    "MaxLength": 40,
    "AllowedPattern": "^[a-zA-z][a-zA-Z0-9]*(-[a-zA-Z0-9]+)*$"
  },
  "InfluxDBUsername": {
    "Description": "The username of the initial admin user created in InfluxDB. Must
start with a letter and can't end with a hyphen or contain two consecutive hyphens.
For example, my-user1. This username will allow you to access the InfluxDB UI to
perform various administrative tasks and also use the InfluxDB CLI to create an
operator token. These attributes will be stored in a Secret created in AWS Secrets
Manager in your account.",
    "Type": "String",
    "Default": "admin",
    "MinLength": 1,
    "MaxLength": 64
  },
  "InfluxDBPassword": {
    "Description": "The password of the initial admin user created in InfluxDB. This
password will allow you to access the InfluxDB UI to perform various administrative
tasks and also use the InfluxDB CLI to create an operator token. These attributes will
be stored in a Secret created in AWS Secrets Manager in your account.",
    "Type": "String",
    "NoEcho": true,
    "MinLength": 8,
    "MaxLength": 64,
    "AllowedPattern": "^[a-zA-Z0-9]+$"
  },
  "InfluxDBOrganization": {

```

```

    "Description": "The name of the initial organization for the initial admin user
in InfluxDB. An InfluxDB organization is a workspace for a group of users.",
    "Type": "String",
    "Default": "org",
    "MinLength": 1,
    "MaxLength": 64
  },
  "InfluxDBBucket": {
    "Description": "The name of the initial InfluxDB bucket. All InfluxDB data
is stored in a bucket. A bucket combines the concept of a database and a retention
period (the duration of time that each data point persists). A bucket belongs to an
organization.",
    "Type": "String",
    "Default": "bucket",
    "MinLength": 2,
    "MaxLength": 64,
    "AllowedPattern": "^[^_\\\\""][^\\\\""]*$"
  },
  "DeploymentType": {
    "Description": "Specifies whether the Timestream for InfluxDB is deployed as
Single-AZ or with a MultiAZ Standby for High availability",
    "Type": "String",
    "Default": "WITH_MULTIAZ_STANDBY",
    "AllowedValues": [
      "SINGLE_AZ",
      "WITH_MULTIAZ_STANDBY"
    ]
  },
  "AllocatedStorage": {
    "Description": "The amount of storage to allocate for your DB storage type in GiB
(gibibytes).",
    "Type": "Number",
    "Default": 400,
    "MinValue": 20,
    "MaxValue": 16384
  },
  "DbInstanceType": {
    "Description": "The Timestream for InfluxDB DB instance type to run InfluxDB
on.",
    "Type": "String",
    "Default": "db.influx.medium",
    "AllowedValues": [
      "db.influx.medium",
      "db.influx.large",

```

```

        "db.influx.xlarge",
        "db.influx.2xlarge",
        "db.influx.4xlarge",
        "db.influx.8xlarge",
        "db.influx.12xlarge",
        "db.influx.16xlarge"
    ]
},
"DbStorageType": {
    "Description": "The Timestream for InfluxDB DB storage type to read and write
InfluxDB data.",
    "Type": "String",
    "Default": "InfluxIOIncludedT1",
    "AllowedValues": [
        "InfluxIOIncludedT1",
        "InfluxIOIncludedT2",
        "InfluxIOIncludedT3"
    ]
},
"PubliclyAccessible": {
    "Description": "Configures the DB instance with a public IP to facilitate
access.",
    "Type": "String",
    "Default": true,
    "AllowedValues": [
        true,
        false
    ]
}
},
"Conditions": {
    "IsMultiAZ": {
        "Fn::Equals": [
            {"Ref": "DeploymentType"},
            "WITH_MULTIAZ_STANDBY"
        ]
    },
    "IsPublic": {
        "Fn::Equals": [
            {"Ref": "PubliclyAccessible"},
            true
        ]
    }
}
},

```

```

"Resources": {
  "VPC": {
    "Type": "AWS::EC2::VPC",
    "Properties": {
      "CidrBlock": {"Ref": "VPCCIDR"}
    }
  },
  "InternetGateway": {
    "Type": "AWS::EC2::InternetGateway",
    "Condition": "IsPublic"
  },
  "InternetGatewayAttachment": {
    "Type": "AWS::EC2::VPCGatewayAttachment",
    "Condition": "IsPublic",
    "Properties": {
      "InternetGatewayId": {"Ref": "InternetGateway"},
      "VpcId": {"Ref": "VPC"}
    }
  },
  "Subnet1": {
    "Type": "AWS::EC2::Subnet",
    "Properties": {
      "VpcId": {"Ref": "VPC"},
      "AvailabilityZone": {
        "Fn::Select": [
          0,
          {"Fn::GetAZs": ""}
        ]
      },
      "CidrBlock": {
        "Fn::Select": [
          0,
          {
            "Fn::Cidr": [
              {
                "Fn::GetAtt": [
                  "VPC",
                  "CidrBlock"
                ]
              },
              2,
              12
            ]
          }
        ]
      }
    }
  }
}

```

```

    ]
  },
  "MapPublicIpOnLaunch": {
    "Fn::If": [
      "IsPublic",
      true,
      false
    ]
  }
},
"Subnet2": {
  "Type": "AWS::EC2::Subnet",
  "Condition": "IsMultiAZ",
  "Properties": {
    "VpcId": {"Ref": "VPC"},
    "AvailabilityZone": {
      "Fn::Select": [
        1,
        {"Fn::GetAZs": ""}
      ]
    },
  },
  "CidrBlock": {
    "Fn::Select": [
      1,
      {
        "Fn::Cidr": [
          {
            "Fn::GetAtt": [
              "VPC",
              "CidrBlock"
            ]
          },
          2,
          12
        ]
      }
    ]
  },
  "MapPublicIpOnLaunch": {
    "Fn::If": [
      "IsPublic",
      true,
      false
    ]
  }
}

```

```

    ]
  }
}
},
"RouteTable": {
  "Type": "AWS::EC2::RouteTable",
  "Properties": {
    "VpcId": {"Ref": "VPC"}
  }
},
"DefaultRoute": {
  "Type": "AWS::EC2::Route",
  "Condition": "IsPublic",
  "DependsOn": "InternetGatewayAttachment",
  "Properties": {
    "RouteTableId": {"Ref": "RouteTable"},
    "DestinationCidrBlock": "0.0.0.0/0",
    "GatewayId": {"Ref": "InternetGateway"}
  }
},
"Subnet1RouteTableAssociation": {
  "Type": "AWS::EC2::SubnetRouteTableAssociation",
  "Properties": {
    "RouteTableId": {"Ref": "RouteTable"},
    "SubnetId": {"Ref": "Subnet1"}
  }
},
"Subnet2RouteTableAssociation": {
  "Type": "AWS::EC2::SubnetRouteTableAssociation",
  "Condition": "IsMultiAZ",
  "Properties": {
    "RouteTableId": {"Ref": "RouteTable"},
    "SubnetId": {"Ref": "Subnet2"}
  }
},
"InfluxDBSecurityGroup": {
  "Type": "AWS::EC2::SecurityGroup",
  "Properties": {
    "GroupName": "influxdb-sg",
    "GroupDescription": "Security group allowing port 8086 ingress for InfluxDB",
    "VpcId": {"Ref": "VPC"}
  }
},
"InfluxDBSecurityGroupIngress": {

```



```

    "Type": "AWS::EC2::SecurityGroupIngress",
    "Properties": {
      "GroupId": {"Ref": "InfluxDBSecurityGroup"},
      "IpProtocol": "tcp",
      "CidrIp": "0.0.0.0/0",
      "FromPort": 8086,
      "ToPort": 8086
    }
  },
  "InfluxDBLogsS3Bucket": {
    "Type": "AWS::S3::Bucket",
    "DeletionPolicy": "Retain"
  },
  "InfluxDBLogsS3BucketPolicy": {
    "Type": "AWS::S3::BucketPolicy",
    "Properties": {
      "Bucket": {"Ref": "InfluxDBLogsS3Bucket"},
      "PolicyDocument": {
        "Version": "2012-10-17",
        "Statement": [
          {
            "Action": "s3:PutObject",
            "Effect": "Allow",
            "Resource": {"Fn::Sub": "arn:aws:s3:::${InfluxDBLogsS3Bucket}/InfluxLogs/
*"},
            "Principal": {"Service": "timestream-influxdb.amazonaws.com"}
          },
          {
            "Action": "s3:*",
            "Effect": "Deny",
            "Resource": [
              {"Fn::Sub": "arn:aws:s3:::${InfluxDBLogsS3Bucket}/*"},
              {"Fn::Sub": "arn:aws:s3:::${InfluxDBLogsS3Bucket}"}
            ],
            "Principal": "*",
            "Condition": {
              "Bool": {"aws:SecureTransport": false}
            }
          }
        ]
      }
    }
  },
  "DbInstance": {

```

```

    "Type": "AWS::Timestream::InfluxDBInstance",
    "DependsOn": "InfluxDBLogsS3BucketPolicy",
    "Properties": {
      "DbStorageType": {"Ref": "DbStorageType"},
      "AllocatedStorage": {"Ref": "AllocatedStorage"},
      "DbInstanceType": {"Ref": "DbInstanceType"},
      "Name": {"Ref": "DbInstanceName"},
      "Username": {"Ref": "InfluxDBUsername"},
      "Password": {"Ref": "InfluxDBPassword"},
      "Organization": {"Ref": "InfluxDBOrganization"},
      "Bucket": {"Ref": "InfluxDBBucket"},
      "PubliclyAccessible": {
        "Fn::If": [
          "IsPublic",
          true,
          false
        ]
      },
      "DeploymentType": {"Ref": "DeploymentType"},
      "VpcSecurityGroupIds": [
        {"Ref": "InfluxDBSecurityGroup"}
      ],
      "VpcSubnetIds": {
        "Fn::If": [
          "IsMultiAZ",
          [
            {"Ref": "Subnet1"},
            {"Ref": "Subnet2"}
          ],
          [
            {"Ref": "Subnet1"}
          ]
        ]
      },
      "LogDeliveryConfiguration": {
        "S3Configuration": {
          "BucketName": {"Ref": "InfluxDBLogsS3Bucket"},
          "Enabled": true
        }
      }
    }
  },
  "Outputs": {

```

```

"VPC": {
  "Description": "A reference to the VPC used to create network resources",
  "Value": {"Ref": "VPC"}
},
"Subnets": {
  "Description": "A list of the subnets created",
  "Value": {
    "Fn::If": [
      "IsMultiAZ",
      {
        "Fn::Join": [
          ",",
          [
            {"Ref": "Subnet1"},
            {"Ref": "Subnet2"}
          ]
        ]
      },
      {"Ref": "Subnet1"}
    ]
  }
},
"Subnet1": {
  "Description": "A reference to the subnet in the 1st Availability Zone",
  "Value": {"Ref": "Subnet1"}
},
"Subnet2": {
  "Condition": "IsMultiAZ",
  "Description": "A reference to the subnet in the 2nd Availability Zone",
  "Value": {"Ref": "Subnet2"}
},
"InfluxDBSecurityGroup": {
  "Description": "Security group with port 8086 ingress rule",
  "Value": {"Ref": "InfluxDBSecurityGroup"}
},
"InfluxDBLogsS3Bucket": {
  "Description": "S3 Bucket containing InfluxDB logs from the DB instance",
  "Value": {"Ref": "InfluxDBLogsS3Bucket"}
},
"DbInstance": {
  "Description": "A reference to the Timestream for InfluxDB DB instance",
  "Value": {"Ref": "DbInstance"}
},
"InfluxAuthParametersSecretArn": {

```

```

    "Description": "The Amazon Resource Name (ARN) of the AWS Secrets Manager secret
    containing the initial InfluxDB authorization parameters. The secret value is a JSON
    formatted key-value pair holding InfluxDB authorization values: organization, bucket,
    username, and password.",
    "Value": {
      "Fn::GetAtt": [
        "DbInstance",
        "InfluxAuthParametersSecretArn"
      ]
    }
  },
  "Endpoint": {
    "Description": "The endpoint URL to connect to InfluxDB",
    "Value": {
      "Fn::Join": [
        "",
        [
          "https://",
          {
            "Fn::GetAtt": [
              "DbInstance",
              "Endpoint"
            ]
          },
          ":8086"
        ]
      ]
    }
  }
}

```

YAML

```

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

```

```

Label:
  default: "Amazon Timestream for InfluxDB Configuration"
Parameters:
  - DbInstanceName
  - InfluxDBUsername
  - InfluxDBPassword
  - InfluxDBOrganization
  - InfluxDBBucket
  - DbInstanceType
  - DbStorageType
  - AllocatedStorage
  - PubliclyAccessible
  - DeploymentType
ParameterLabels:
  VPCCIDR:
    default: VPC CIDR

```

Parameters:

Network Configuration

VPCCIDR:

Description: Please enter the IP range (CIDR notation) for the new VPC

Type: String

Default: 10.0.0.0/16

Timestream for InfluxDB Configuration

DbInstanceName:

Description: The name that uniquely identifies the DB instance when interacting with the Amazon Timestream for InfluxDB API and CLI commands. This name will also be a prefix included in the endpoint. DB instance names must be unique per customer and per Region.

Type: String

Default: mydbinstance

MinLength: 3

MaxLength: 40

AllowedPattern: ^[a-zA-z][a-zA-Z0-9]*(-[a-zA-Z0-9]+)*\$

InfluxDB initial user configurations

InfluxDBUsername:

Description: The username of the initial admin user created in InfluxDB. Must start with a letter and can't end with a hyphen or contain two consecutive hyphens. For example, my-user1. This username will allow you to access the InfluxDB UI to perform various administrative tasks and also use the InfluxDB CLI to create an operator token. These attributes will be stored in a Secret created in AWS Secrets Manager in your account.

Type: String

Default: admin

MinLength: 1

MaxLength: 64

InfluxDBPassword:

Description: The password of the initial admin user created in InfluxDB. This password will allow you to access the InfluxDB UI to perform various administrative tasks and also use the InfluxDB CLI to create an operator token. These attributes will be stored in a Secret created in AWS in your account.

Type: String

NoEcho: true

MinLength: 8

MaxLength: 64

AllowedPattern: ^[a-zA-Z0-9]+\$

InfluxDBOrganization:

Description: The name of the initial organization for the initial admin user in InfluxDB. An InfluxDB organization is a workspace for a group of users.

Type: String

Default: org

MinLength: 1

MaxLength: 64

InfluxDBBucket:

Description: The name of the initial InfluxDB bucket. All InfluxDB data is stored in a bucket. A bucket combines the concept of a database and a retention period (the duration of time that each data point persists). A bucket belongs to an organization.

Type: String

Default: bucket

MinLength: 2

MaxLength: 64

AllowedPattern: ^[^\"]*\"

DeploymentType:

Description: Specifies whether the Timestream for InfluxDB is deployed as Single-AZ or with a MultiAZ Standby for High availability

Type: String

Default: WITH_MULTIAZ_STANDBY

AllowedValues:

- SINGLE_AZ

- WITH_MULTIAZ_STANDBY

AllocatedStorage:

Description: The amount of storage to allocate for your DB storage type in GiB (gibibytes).

Type: Number

Default: 400

MinValue: 20

MaxValue: 16384

DbInstanceType:

Description: The Timestream for InfluxDB DB instance type to run InfluxDB on.

Type: String

Default: db.influx.medium

AllowedValues:

- db.influx.medium
- db.influx.large
- db.influx.xlarge
- db.influx.2xlarge
- db.influx.4xlarge
- db.influx.8xlarge
- db.influx.12xlarge
- db.influx.16xlarge

DbStorageType:

Description: The Timestream for InfluxDB DB storage type to read and write InfluxDB data.

Type: String

Default: InfluxIOIncludedT1

AllowedValues:

- InfluxIOIncludedT1
- InfluxIOIncludedT2
- InfluxIOIncludedT3

PubliclyAccessible:

Description: Configures the DB instance with a public IP to facilitate access.

Type: String

Default: true

AllowedValues:

- true
- false

Conditions:

IsMultiAZ: !Equals [!Ref DeploymentType, WITH_MULTIAZ_STANDBY]

IsPublic: !Equals [!Ref PubliclyAccessible, true]

Resources:

VPC:

Type: AWS::EC2::VPC

Properties:

CidrBlock: !Ref VPCCIDR

InternetGateway:

Type: AWS::EC2::InternetGateway

Condition: IsPublic

InternetGatewayAttachment:

Type: AWS::EC2::VPCElasticNetworkInterfaceAttachment

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"]]
```

Implementación de pilas basadas en Windows usando CloudFormation

Esta página proporciona enlaces a la documentación de referencia técnica para los recursos de CloudFormation que se utilizan habitualmente en las implementaciones basadas en Windows.

CloudFormation proporciona soporte para implementar y administrar pilas de Microsoft Windows a través de Infrastructure as Code (IaC). Puede usar CloudFormation para el aprovisionamiento automatizado de instancias de EC2 basadas en Windows, SQL Server en Amazon RDS y el Active Directory de Microsoft a través de Directory Service.

AWS proporciona imágenes de máquina de Amazon (AMI) preconfiguradas diseñadas específicamente para plataformas Windows que le ayudan a implementar aplicaciones rápidamente en Amazon EC2. Estas AMI incluyen configuraciones predeterminadas de Microsoft y personalizaciones específicas de AWS. Con CloudFormation, puede elegir una AMI adecuada, inicializar una instancia y acceder a ella utilizando Conexión a Escritorio remoto, como lo haría con cualquier otro Windows Server. Las AMI contienen componentes de software esenciales, incluyendo EC2Launch (las versiones varían según la edición de Windows Server), AWS Systems Manager, CloudFormation, Herramientas de AWS para PowerShell y varios controladores de red, almacenamiento y gráficos para garantizar un rendimiento y una compatibilidad óptimos con los servicios de AWS. Para obtener más información, consulte la [Referencia de la AMI de Windows de AWS](#).

CloudFormation también admite herramientas de configuración de software, como scripts UserData, que pueden ejecutar comandos PowerShell o por lotes cuando se inicia una instancia de EC2 por

primera vez. También ofrece scripts auxiliares (`cfn-init`, `cfn-signal`, `cfn-get-metadata` y `cfn-hup`) y admite los metadatos `AWS::CloudFormation::Init` para administrar paquetes, archivos y servicios en instancias de Windows.

Para entornos empresariales, CloudFormation permite la unión de dominios, la administración de licencias de Windows mediante modelos de licencias de EC2 y la gestión segura de credenciales con AWS Secrets Manager. En combinación con plantillas con control de versiones e implementaciones repetibles, CloudFormation ayuda a las organizaciones a mantener entornos de Windows consistentes, seguros y escalables en múltiples cuentas y Regiones de AWS.

Para obtener información detallada sobre los recursos de CloudFormation que se utilizan habitualmente en las implementaciones basadas en Windows, consulte los siguientes temas de referencia técnica.

Tipo de recurso	Descripción
AWS::EC2::Instance	Para inicializar instancias de EC2 en Windows.
AWS::EC2::SecurityGroup	Para definir las reglas de firewall para cargas de trabajo de Windows.
AWS::AutoScaling::AutoScalingGroup	Para escalar instancias de EC2 en Windows.
AWS::EC2::LaunchTemplate	
AWS::DirectoryService::MicrosoftAD	Para implementar el Active Directory de Microsoft.
AWS::FSx::FileSystem	Para implementar FSx para Windows File Server.
AWS::RDS::DBInstance	Para aprovisionar SQL Server en Amazon RDS.
AWS::CloudFormation::Init	Se utiliza en los metadatos de EC2 para configurar instancias. Para obtener más información, consulte Arranque de pilas de CloudFormation basadas en Windows .
AWS::SecretsManager::Secret	Para administrar de forma segura las credenciales y contraseñas de Windows.

Tipo de recurso	Descripción
AWS::SSM::Parameter	Para almacenar los valores de configuración de forma segura.
AWS::IAM::InstanceProfile AWS::IAM::Role	Para conceder permisos a aplicaciones que se ejecutan en instancias de EC2.

Arranque de pilas de CloudFormation basadas en Windows

En este tema se describe cómo arrancar una pila de Windows y solucionar problemas de creación de pila.

Temas

- [Datos de usuario en instancias de EC2](#)
- [Scripts auxiliares de CloudFormation](#)
- [Ejemplo de arranque de una pila de Windows](#)
- [Evite las barras invertidas en las rutas de los archivos de Windows](#)
- [Administración de servicios de Windows](#)
- [Cómo solucionar problemas de creación de pila](#)

Datos de usuario en instancias de EC2

Los datos de usuario son una característica de Amazon EC2 que le permite pasar scripts o información de configuración a una instancia de EC2 cuando se lanza.

En el caso de las instancias de EC2 de Windows:

- Puede utilizar scripts por lotes (mediante etiquetas <script>) o scripts PowerShell (mediante etiquetas <powershell>).
- EC2Launch gestiona la ejecución del script.

Important

Si va a crear su propia AMI de Windows para usarla con CloudFormation, asegúrese de que EC2Launch v2 esté configurada correctamente. EC2Launch v2 es necesario para que

las herramientas de arranque de CloudFormation inicialicen y configuren correctamente las instancias de Windows durante la creación de la pila. Para obtener más información, consulte [Uso del agente EC2Launch v2 para realizar tareas durante la inicialización de una instancia de EC2 Windows](#) en la Guía del usuario de Amazon EC2.

Para obtener información sobre las AMI de Windows de AWS, consulte [AWS Windows AMI Reference](#).

Scripts auxiliares de CloudFormation

Los scripts auxiliares son utilidades para configurar instancias durante el proceso de arranque. Se utilizan con los datos de usuario de Amazon EC2 y ofrecen opciones eficaces de configuración.

CloudFormation proporciona los siguientes scripts auxiliares de Python que puede utilizar para instalar software e iniciar servicios en una instancia de Amazon EC2 que cree como parte de la pila:

- `cfn-init`: se utiliza para recuperar e interpretar los metadatos de los recursos, instalar paquetes, crear archivos e iniciar servicios.
- `cfn-signal`: se utiliza para señalar con una `CreationPolicy`, de modo que se puedan sincronizar otros recursos de la pila cuando esté listo el recurso o la aplicación de requisitos previos.
- `cfn-get-metadata`: se utiliza para recuperar los metadatos de un recurso o de la ruta a una clave específica.
- `cfn-hup`: se utiliza para comprobar si hay actualizaciones de metadatos y ejecutar enlaces personalizados cuando se detectan cambios.

Los scripts se llaman directamente desde la plantilla. Los scripts funcionan junto con los metadatos de los recursos que están definidos en la misma plantilla. Los scripts se ejecutan en la instancia Amazon EC2 durante el proceso de creación de la pila.

Para obtener más información, consulte [Referencia de scripts auxiliares de CloudFormation](#) en la Guía de referencia de plantillas de CloudFormation.

Ejemplo de arranque de una pila de Windows

Examinemos fragmentos de ejemplo de una plantilla de Windows Server que realiza las siguientes acciones:

- Lanza una instancia de EC2 con el nombre `TestInstance` de una AMI de Windows Server de 2022.
- Crea un archivo de prueba sencillo para verificar que `cfn-init` está funcionando.
- Configura `cfn-hup` para la administración continua de la configuración.
- Utiliza una `CreationPolicy` para garantizar que la instancia señale la finalización exitosa.

El script auxiliar `cfn-init` se utiliza para llevar a cabo cada una de estas acciones en función de la información del recurso `AWS::CloudFormation::Init` de la plantilla.

La sección `AWS::CloudFormation::Init` se denomina `TestInstance` y comienza con la siguiente instrucción.

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

Después de esto, la sección `files` de `AWS::CloudFormation::Init` se declara.

```
create_files:
  files:
    c:\cfn\test.txt:
      content: !Sub |
        Hello from ${AWS::StackName}
    c:\cfn\cfn-hup.conf:
      content: !Sub |
        [main]
        stack=${AWS::StackName}
        region=${AWS::Region}
        interval=2
    c:\cfn\hooks.d\cfn-auto-reloader.conf:
      content: !Sub |
        [cfn-auto-reloader-hook]
        triggers=post.update
        path=Resources.TestInstance.Metadata.AWS::CloudFormation::Init
```

```
    action=cfn-init.exe -v -s ${AWS::StackName} -r TestInstance -c default --
    region ${AWS::Region}
```

Aquí se crean tres archivos, que se colocan en el directorio C:\cfn de la instancia del servidor:

- `test.txt`, un archivo de prueba simple que verifica que `cfn-init` funciona correctamente y puede crear archivos con contenido dinámico.
- `cfn-hup.conf`, el archivo de configuración para `cfn-hup` con un intervalo de verificación de 2 minutos.
- `cfn-auto-reloader.conf`, el archivo de configuración para el enlace que usa `cfn-hup` para iniciar una actualización (mediante llamada a `cfn-init`) cuando cambian los metadatos en `AWS::CloudFormation::Init`.

La siguiente es la sección `start_services` que configura los servicios de Windows.

```
start_services:
  services:
    windows:
      cfn-hup:
        enabled: true
        ensureRunning: true
        files:
          - c:\cfn\cfn-hup.conf
          - c:\cfn\hooks.d\cfn-auto-reloader.conf
```

Esta sección garantiza que el servicio `cfn-hup` se inicie y se reinicie automáticamente si se modifican los archivos de configuración. El servicio supervisa los cambios en los metadatos de CloudFormation y vuelve a ejecutar `cfn-init` cuando se detectan actualizaciones.

A continuación está la sección `Properties`.

```
TestInstance:
  Type: AWS::EC2::Instance
  CreationPolicy:
    ResourceSignal:
      Timeout: PT20M
  Metadata:
    AWS::CloudFormation::Init:
      # ... metadata configuration ...
```



```
Properties:
  InstanceType: t2.large
  ImageId: '{{resolve:ssm:/aws/service/ami-windows-latest/Windows_Server-2022-
English-Full-Base}}'
  SecurityGroupIds:
    - !Ref InstanceSecurityGroup
  KeyName: !Ref KeyPairName
  UserData:
    Fn::Base64: !Sub |
      <powershell>
        cfn-init.exe -v -s ${AWS::StackName} -r TestInstance -c default --region
${AWS::Region}
        cfn-signal.exe -e $lastexitcode --stack ${AWS::StackName} --resource
TestInstance --region ${AWS::Region}
      </powershell>
```

En esta sección, la propiedad `UserData` contiene un script PowerShell que ejecutará `EC2Launch`, rodeado de etiquetas `<powershell>`. El script ejecuta `cfn-init` con el ConfigSet `default` y luego utiliza `cfn-signal` para informar el código de salida a CloudFormation. La `CreationPolicy` se utiliza para garantizar que la instancia está configurada correctamente antes de que se dé por finalizada la creación de la pila.

La propiedad `ImageId` utiliza un parámetro público de Systems Manager Parameter Store para recuperar automáticamente el último ID de AMI del Windows Server de 2022. Este enfoque elimina la necesidad de mapeos de AMI específicos de la región y garantiza que usted siempre obtenga la AMI más reciente. El uso de parámetros de Systems Manager para los ID de AMI es una práctica recomendada para mantener las referencias de AMI actuales. Si planea conectarse a su instancia, asegúrese de que la propiedad `SecurityGroupIds` haga referencia a un grupo de seguridad que permita el acceso RDP.

La `CreationPolicy` se declara como parte de las propiedades del recurso y especifica un periodo de tiempo de espera. El comando `cfn-signal` en las señales de datos de usuario indica cuándo se ha completado la configuración de la instancia:

```
TestInstance:
  Type: AWS::EC2::Instance
  CreationPolicy:
    ResourceSignal:
      Timeout: PT20M
  Properties:
    # ... other properties ...
```

Como el proceso de arranque es mínimo y solo crea archivos e inicia los servicios, la `CreationPolicy` espera 20 minutos (PT20M) antes de que se agote el tiempo de espera. El tiempo de espera se especifica mediante el formato de duración ISO 8601. Tenga en cuenta que las instancias de Windows suelen tardar más en lanzarse que las instancias de Linux, por lo que debe realizar pruebas exhaustivas para determinar los valores de tiempo de espera más adecuados para sus necesidades.

Si todo va bien, la `CreationPolicy` se completa correctamente y usted puede acceder a la instancia del Windows Server mediante su dirección IP pública. Una vez completada la creación de la pila, el ID de la instancia y la dirección IP aparecerán en la pestaña Salidas de la consola de CloudFormation.

Outputs:

```
InstanceId:
  Value: !Ref TestInstance
  Description: Instance ID of the Windows Server
PublicIP:
  Value: !GetAtt TestInstance.PublicIp
  Description: Public IP address of the Windows Server
```

También puede comprobar manualmente que el arranque ha funcionado correctamente conectándose a la instancia mediante RDP y comprobando que el archivo `C:\cfn\test.txt` existe y contiene el contenido esperado. Para obtener información acerca de cómo conectarse a una instancias de Windows, consulte [Conexión a la instancia de Windows mediante RDP](#) en la Guía del usuario de Amazon EC2.

Evite las barras invertidas en las rutas de los archivos de Windows

Al hacer referencia a las rutas de Windows en las plantillas de CloudFormation, recuerde siempre evitar correctamente las barras invertidas (`\`) según el formato de plantilla que utilice.

- En el caso de las plantillas JSON, debe usar barras invertidas dobles en las rutas de los archivos de Windows, ya que JSON trata a la barra invertida como un carácter de escape. La primera barra invertida escapa a la segunda, lo que da como resultado la interpretación de una sola barra invertida literal.

```
"commands" : {
  "1-extract" : {
    "command" : "C:\\\\SharePoint\\SharePointFoundation2010.exe /extract:C:\\\\SharePoint
\\\\SPF2010 /quiet /log:C:\\\\SharePoint\\SharePointFoundation2010-extract.log"
```

```
}  
}
```

- En el caso de las plantillas YAML, las barras invertidas únicas suelen ser suficientes.

```
commands:  
  1-extract:  
    command: C:\SharePoint\SharePointFoundation2010.exe /extract:C:\SharePoint  
             \SPF2010 /quiet /log:C:\SharePoint\SharePointFoundation2010-extract.log
```

Administración de servicios de Windows

Puede administrar los servicios de Windows de la misma forma que los servicios de Linux, salvo que use una clave `windows` en lugar de `sysvinit`. El siguiente ejemplo inicia el servicio `cfn-hup`, lo establece en automático y reinicia el servicio si `cfn-init` modifica los archivos de configuración `c:\cfn\cfn-hup.conf` o `c:\cfn\hooks.d\cfn-auto-reloader.conf`.

```
services:  
  windows:  
    cfn-hup:  
      enabled: true  
      ensureRunning: true  
      files:  
        - c:\cfn\cfn-hup.conf  
        - c:\cfn\hooks.d\cfn-auto-reloader.conf
```

Puede administrar otros servicios de Windows de la misma forma mediante el nombre, no el nombre de visualización, para hacer referencia al servicio.

Cómo solucionar problemas de creación de pila

Si la pila falla durante la creación, el comportamiento predeterminado es de restauración en caso de error. Mientras que normalmente dicho comportamiento es bueno, ya que evita gastos innecesarios, dificulta la depuración de por qué falla la creación de la pila.

Para desactivar este comportamiento al crear o actualizar la pila con la consola de CloudFormation, seleccione la opción **Preservar los recursos aprovisionados correctamente** en **Opciones de error de pila**. Para obtener más información, consulte [Elija cómo gestionar los errores al aprovisionar recursos](#). Esto le permite iniciar sesión en la instancia y ver el archivo de registros para identificar los problemas encontrados al ejecutar sus scripts de arranque.

Es importante fijarse en estos registros:

- El registro de configuración de EC2 en %ProgramData%\Amazon\EC2Launch\log\agent.log
- El registro cfn-init en C:\cfn\log\cfn-init.log (comprueba los códigos de salida y los mensajes de error para ver si hay puntos de fallo específicos)

Para obtener más registros, consulte los siguientes temas en la Guía del usuario de Amazon EC2:

- [EC2Launch Estructura de directorios de](#)
- [EC2Launch Estructura de directorios de v](#)

Para obtener más información sobre la solución de problemas de arranque, consulte [¿Cómo se solucionan los problemas de los scripts auxiliares que no se inician en una pila de CloudFormation con instancias de Windows?](#).

Amplíe las capacidades de su plantilla con los tipos de recursos proporcionados por CloudFormation

CloudFormation ofrece varios tipos de recursos que puede usar en las plantillas de pila para ampliar sus capacidades más allá de las de una plantilla de pila simple.

Estos tipos de recursos incluyen:

Tipo de recurso	Descripción	Documentación
Recursos personalizados	El tipo de recurso <code>AWS::CloudFormation::CustomResource</code> permite crear recursos personalizados capaces de realizar tareas de aprovisionamiento específicas o incluir recursos que no están disponibles como tipos de recursos de CloudFormation.	Recursos personalizados
Macros	El tipo de recurso <code>AWS::CloudFormation::Macro</code> define una pieza de código reutilizable capaz de realizar procesamientos personalizados en plantillas de CloudForm	Macros de plantillas

Tipo de recurso	Descripción	Documentación
	ation. Las macros pueden modificar las plantillas, generar recursos adicionales o realizar otras operaciones personalizadas durante la creación o las actualizaciones de las pilas.	
Pilas anidadas	El tipo de recurso <code>AWS::CloudFormation::Stack</code> permite crear pilas anidadas en las plantillas de CloudFormation para obtener arquitecturas de pilas más modulares y reutilizables.	Pilas anidadas
StackSet	El tipo de recurso <code>AWS::CloudFormation::StackSet</code> crea o actualiza un StackSet de CloudFormation, que es un contenedor de pilas que se pueden implementar en varias Cuentas de AWS y regiones.	Administración de pilas con StackSets
Condición de espera	El tipo de recurso <code>AWS::CloudFormation::WaitCondition</code> detiene la creación o actualización de la pila hasta que se cumpla una condición específica, como la finalización satisfactoria de un proceso prolongado o la disponibilidad de recursos externos.	Condiciones de espera
Identificador de condición de espera	El tipo de recurso <code>AWS::CloudFormation::WaitConditionHandle</code> funciona conjuntamente con el tipo de recurso <code>AWS::CloudFormation::WaitCondition</code> . Proporciona una URL prefirmada que se utiliza para enviar señales que indican que se ha cumplido una condición específica. Estas señales permiten continuar con el proceso de creación o actualización de la pila.	Condiciones de espera

Crear una lógica de aprovisionamiento personalizada con recursos personalizados

Los recursos personalizados le permiten escribir la lógica de aprovisionamiento personalizada en sus plantillas de CloudFormation y hacer que CloudFormation la ejecute cada vez que cree, actualice (si ha cambiado el recurso personalizado) o elimine una pila. Esto puede resultar útil cuando sus requisitos de aprovisionamiento implican una lógica compleja o flujos de trabajo que no se pueden expresar con los tipos de recursos integrados de CloudFormation.

Por ejemplo, es posible que desee incluir recursos que no están disponibles como tipos de recursos de CloudFormation. Puede incluir esos recursos usando recursos personalizados. De esta forma, puede seguir administrando todos los recursos relacionados en una sola pila.

Para definir un recurso personalizado en su plantilla de CloudFormation, se usan los tipos de recurso `AWS::CloudFormation::CustomResource` o `Custom::MyCustomResourceTypeName`. Los recursos personalizados requieren una propiedad, el token de servicio, que especifica a dónde envía CloudFormation las solicitudes, como un tema de Amazon SNS o una función de Lambda.

En los siguientes temas se proporciona información sobre cómo utilizar recursos personalizados.

Temas

- [Cómo funcionan los recursos personalizados](#)
- [Tiempo de espera de respuesta](#)
- [Referencia de solicitud y respuesta de recursos personalizados de CloudFormation](#)
- [Recursos personalizados y respaldados por Amazon SNS](#)
- [Recursos personalizados y respaldados por Lambda](#)

Note

El registro de CloudFormation y los recursos personalizados ofrecen sus propias ventajas. Los recursos personalizados ofrecen las siguientes ventajas:

- No es necesario registrar el recurso.
- Puede incluir un recurso completo como parte de una plantilla sin necesidad de registrarse.
- Son compatibles con las operaciones Create, Update y Delete.

Entre las ventajas de los recursos basados en el registro se encuentran:

- Compatibilidad con el modelado, el aprovisionamiento y la administración de recursos de aplicaciones de terceros
- Compatibilidad con las operaciones Create, Read, Update, Delete y List (CRUDL)
- Compatibilidad con la detección de desviaciones en tipos de recursos privados y de terceros

A diferencia de los recursos personalizados, los recursos basados en registros no necesitarán asociar un tema de Amazon SNS o una función de Lambda para realizar operaciones CRUDL. Para obtener más información, consulte [Administración de extensiones a través del registro de CloudFormation](#).

Cómo funcionan los recursos personalizados

El proceso general para configurar un nuevo recurso personalizado incluye los siguientes pasos. Estos pasos implican dos funciones: el proveedor de recursos personalizados, que es el propietario del recurso personalizado, y el desarrollador de plantillas, que crea una plantilla que incluye un tipo de recurso personalizado. Puede ser la misma persona, pero si no lo es, el proveedor de recursos personalizados debe colaborar con el desarrollador de plantillas.

1. El proveedor de recursos personalizados escribe una lógica que determina cómo gestionar las solicitudes de CloudFormation y llevar a cabo acciones en el recurso personalizado.
2. El proveedor de recursos personalizados crea el tema de Amazon SNS o la función de Lambda a la que CloudFormation puede enviar solicitudes. El tema de Amazon SNS o la función de Lambda deben estar en la misma región en la que se creará la pila.
3. El proveedor de recursos personalizados proporciona el ARN del tema de Amazon SNS o el ARN de la función de Lambda al desarrollador de plantillas.
4. El desarrollador de plantillas define el recurso personalizado en su plantilla de CloudFormation. Esto incluye un token de servicio y los parámetros de datos de entrada. El proveedor de recursos personalizados define el token de servicio y la estructura de los datos de entrada. El token de servicio, que siempre es obligatorio, especifica el ARN del tema de Amazon SNS o el ARN de la función de Lambda. Los datos de entrada son opcionales según el recurso personalizado.

Ahora, cada vez que se utiliza la plantilla para crear, actualizar o eliminar el recurso personalizado, CloudFormation envía una solicitud al token de servicio especificado y espera una respuesta antes de proceder con la operación de la pila.

A continuación, se resume el flujo de creación de una pila a partir de la plantilla:

1. CloudFormation envía una solicitud al token de servicio especificado. La solicitud incluye información como el tipo de solicitud y una URL prefirmada de bucket de Amazon S3, donde el recurso personalizado envía sus respuestas. Para obtener más información acerca de lo que se incluye en la solicitud, consulte [Referencia de solicitud y respuesta de recursos personalizados de CloudFormation](#).

A continuación, se presenta un ejemplo de los datos que CloudFormation incluye en una solicitud de Create: En este ejemplo, ResourceProperties permite a CloudFormation crear una carga personalizada que se enviará a la función de Lambda.

```
{
  "RequestType" : "Create",
  "RequestId" : "unique id for this create request",
  "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-
cd98-11ea-90d5-0a9cd3354c10",
  "ResponseURL" : "http://pre-signed-S3-url-for-response",
  "ResourceType" : "Custom::TestResource",
  "LogicalResourceId" : "MyTestResource",
  "ResourceProperties" : {
    "Name" : "Value",
    "List" : [ "1", "2", "3" ]
  }
}
```

2. El proveedor de recursos personalizados procesa la solicitud de CloudFormation y devuelve una respuesta SUCCESS o FAILED a la URL prefirmada. custom resource provider proporciona la respuesta en un archivo con formato JSON y la carga a la URL de S3 prefirmada. Para obtener más información, consulte [Carga de objetos con URL prefirmadas](#) en la Guía para usuarios de Amazon Simple Storage Service.

En la respuesta, el custom resource provider también puede incluir pares de nombre-valor a los que el template developer puede obtener acceso. Por ejemplo, la respuesta puede incluir datos de salida si la solicitud ha funcionado o un mensaje de error si ha fallado. Para obtener más

información acerca de las respuestas, consulte [Referencia de solicitud y respuesta de recursos personalizados de CloudFormation](#).

 Important

Si los pares nombre-valor contienen información confidencial, debe usar el campo NoEcho para enmascarar la salida del recurso personalizado. De lo contrario, los valores son visibles a través de las API que muestran valores de propiedades (como DescribeStackEvents).

Para obtener más información sobre el uso de NoEcho para enmascarar información confidencial, consulte la práctica recomendada de [No integre credenciales en sus plantillas](#).

custom resource provider es responsable de escuchar y responder a la solicitud. Por ejemplo, en el caso de las notificaciones de Amazon SNS, el proveedor de recursos personalizados debe escuchar las notificaciones que se envían al ARN de un tema específico y responder a ellas. CloudFormation espera y escucha una respuesta en la ubicación de la URL prefirmada.

Los siguientes datos de ejemplo muestran lo que un recurso personalizado podría incluir en una respuesta:

```
{
  "Status" : "SUCCESS",
  "RequestId" : "unique id for this create request",
  "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-cd98-11ea-90d5-0a9cd3354c10",
  "LogicalResourceId" : "MyTestResource",
  "PhysicalResourceId" : "TestResource1",
  "Data" : {
    "OutputName1" : "Value1",
    "OutputName2" : "Value2",
  }
}
```

- Después de obtener una respuesta SUCCESS, CloudFormation continúa con la operación de la pila. Si se recibe una respuesta FAILED o ninguna respuesta, la operación genera un error. Todo dato de salida del recurso personalizado se almacena en la ubicación de la URL prefirmada. El desarrollador de plantillas puede recuperar los datos usando la función [Fn::GetAtt](#).

 Note

Si utiliza AWS PrivateLink, los recursos personalizados de la VPC deben tener acceso a buckets de S3 específicos de CloudFormation. Los recursos personalizados deben enviar las respuestas a una URL de Amazon S3 prefirmada. Si no pueden enviar respuestas a Amazon S3, CloudFormation no recibirá una respuesta y la operación de pila producirá un error. Para obtener más información, consulte [Acceso a CloudFormation a través de un punto de conexión de la interfaz \(AWS PrivateLink\)](#).

Tiempo de espera de respuesta

El tiempo de espera predeterminado para un recurso personalizado es de 3600 segundos (1 hora). Si no se recibe una respuesta en este tiempo, la operación de la pila se considerará fallida.

Puede ajustar el valor del tiempo de espera en función del tiempo que estime que tomará la respuesta del recurso personalizado. Por ejemplo, al aprovisionar un recurso personalizado que invoca una función de Lambda y se espera que responda en cinco minutos, puede establecer un tiempo de espera de cinco minutos en la plantilla de la pila al especificar la propiedad `ServiceTimeout`. Para obtener más información, consulte [Referencia de solicitud y respuesta de recursos personalizados de CloudFormation](#). De esta forma, si hay un error en la función de Lambda que provoca que se bloquee, CloudFormation no podrá realizar la operación de la pila después de cinco minutos, en lugar de esperar toda una hora.

Sin embargo, procure no establecer un valor de tiempo de espera demasiado bajo. Para evitar tiempos de espera inesperados, asegúrese de que su recurso personalizado disponga de tiempo suficiente para completar las acciones necesarias y devolver una respuesta.

Referencia de solicitud y respuesta de recursos personalizados de CloudFormation

CloudFormation administra los recursos personalizados mediante un protocolo de solicitud-respuesta que se comunica con su proveedor de recursos personalizados. Cada solicitud incluye un tipo de solicitud (`Create`, `Update` o `Delete`) y sigue este flujo de trabajo de alto nivel:

1. Un desarrollador de plantillas define un recurso personalizado con un `ServiceToken` y `ServiceTimeout` en la plantilla e inicia una operación de pila.
2. CloudFormation envía una solicitud JSON al proveedor de recursos personalizados a través de SNS o Lambda.

3. El proveedor de recursos personalizados procesa la solicitud y devuelve una respuesta JSON a una URL de bucket de Amazon S3 prefirmaada antes de que venza el tiempo de espera.
4. CloudFormation lee la respuesta y continúa con la operación de la pila. Si no se recibe respuesta alguna antes de que finalice el tiempo de espera, se considera que la solicitud no se ha realizado en la forma correcta y la operación de la pila fallará.

Para obtener más información, consulte [Cómo funcionan los recursos personalizados](#).

En esta sección se describen la estructura, los parámetros y las respuestas esperadas para cada tipo de solicitud.

 Note

El tamaño total del cuerpo de la respuesta no puede superar los 4096 bytes.

Configuración de la plantilla

Al definir un recurso personalizado en una plantilla, el desarrollador de la plantilla utiliza [AWS::CloudFormation::CustomResource](#) con las siguientes propiedades:

ServiceToken

Un ARN de tema de Amazon SNS o un ARN de función de Lambda de la misma región que la pila.

Obligatorio: sí

Tipo: cadena

ServiceTimeout

El tiempo máximo, en segundos, antes de que una operación de recurso personalizado agote el tiempo de espera. Este valor debe estar entre 1 y 3600. Valor predeterminado: 3600 segundos (1 hora).

Obligatorio: no

Tipo: cadena

Se admiten las propiedades del recurso adicional. Las propiedades de los recursos se incluirán como `ResourceProperties` en la solicitud. El proveedor de recursos personalizados debe determinar qué propiedades son válidas y sus valores aceptables.

Objeto de solicitud

Create

Cuando el desarrollador de la plantilla crea una pila que contiene un recurso personalizado, CloudFormation envía una solicitud con `RequestType` establecido en `Create`.

Cree solicitudes que contengan los siguientes campos:

`RequestType`

`Create`.

Obligatorio: sí

Tipo: cadena

`RequestId`

Un ID único para la solicitud.

La combinación de `StackId` con `RequestId` forma un valor que puede usar para identificar de forma única una solicitud en un recurso personalizado en particular.

Obligatorio: sí

Tipo: cadena

`StackId`

El nombre de recurso de Amazon (ARN) que identifica la pila que contiene el recurso personalizado.

La combinación de `StackId` con `RequestId` forma un valor que puede usar para identificar de forma única una solicitud en un recurso personalizado en particular.

Obligatorio: sí

Tipo: cadena

ResponseURL

La URL de respuesta identifica un bucket de S3 prefirmado que recibe respuestas del proveedor de recursos personalizados para CloudFormation.

Obligatorio: sí

Tipo: cadena

ResourceType

El tipo de recurso elegido por el desarrollador de plantillas del recurso personalizado en la plantilla de CloudFormation. Los nombres de tipos de recursos personalizados pueden tener una longitud máxima de 60 caracteres y pueden incluir caracteres alfanuméricos y los siguientes caracteres: `_@-`.

Obligatorio: sí

Tipo: cadena

LogicalResourceId

El nombre elegido por el desarrollador de la plantilla (ID lógico) del recurso personalizado en la plantilla de CloudFormation.

Obligatorio: sí

Tipo: cadena

ResourceProperties

Este campo contiene el contenido del objeto `Properties` enviado por el desarrollador de la plantilla. El proveedor de recursos personalizado define su contenido.

Obligatorio: no

Tipo: objeto JSON

Ejemplo

```
{
  "RequestType" : "Create",
  "RequestId" : "unique-request-id",
  "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/id",
  "ResponseURL" : "pre-signed-url-for-create-response",
```

```
"ResourceType" : "Custom::MyCustomResourceType",
"LogicalResourceId" : "resource-logical-id",
"ResourceProperties" : {
  "key1" : "string",
  "key2" : [ "list" ],
  "key3" : { "key4" : "map" }
}
```

Update

Cuando el desarrollador de la plantilla realiza cambios en las propiedades de un recurso personalizado dentro de la plantilla y actualiza la pila, CloudFormation envía una solicitud al proveedor del recurso personalizado con `RequestType` establecido en `Update`. Esto significa que el código de su recurso personalizado no tiene que detectar cambios en los recursos, porque sabe que sus propiedades cambiaron cuando el tipo de solicitud es `Update`.

Actualice solicitudes que contengan los siguientes campos:

RequestType

`Update`.

Obligatorio: sí

Tipo: cadena

RequestId

Un ID único para la solicitud.

La combinación de `StackId` con `RequestId` forma un valor que puede usar para identificar de forma única una solicitud en un recurso personalizado en particular.

Obligatorio: sí

Tipo: cadena

StackId

El nombre de recurso de Amazon (ARN) que identifica la pila que contiene el recurso personalizado.

La combinación de `StackId` con `RequestId` forma un valor que puede usar para identificar de forma única una solicitud en un recurso personalizado en particular.

Obligatorio: sí

Tipo: cadena

ResponseURL

La URL de respuesta identifica un bucket de S3 prefirmado que recibe respuestas del proveedor de recursos personalizados para CloudFormation.

Obligatorio: sí

Tipo: cadena

ResourceType

El tipo de recurso elegido por el desarrollador de plantillas del recurso personalizado en la plantilla de CloudFormation. Los nombres de tipos de recursos personalizados pueden tener una longitud máxima de 60 caracteres y pueden incluir caracteres alfanuméricos y los siguientes caracteres: `_@-`. No puede cambiar el tipo durante una actualización.

Obligatorio: sí

Tipo: cadena

LogicalResourceId

El nombre elegido por el desarrollador de la plantilla (ID lógico) del recurso personalizado en la plantilla de CloudFormation.

Obligatorio: sí

Tipo: cadena

PhysicalResourceId

Un ID físico definido por el proveedor de recursos personalizados que es exclusivo de dicho proveedor.

Obligatorio: sí

Tipo: cadena

ResourceProperties

Este campo contiene el contenido del objeto `Properties` enviado por el desarrollador de la plantilla. El proveedor de recursos personalizado define su contenido.

Obligatorio: no

Tipo: objeto JSON

OldResourceProperties

Usado solo para solicitudes Update. Los valores de propiedades de recursos declarados con anterioridad por el desarrollador de plantillas en la plantilla de CloudFormation.

Obligatorio: sí

Tipo: objeto JSON

Ejemplo

```
{
  "RequestType" : "Update",
  "RequestId" : "unique-request-id",
  "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/id",
  "ResponseURL" : "pre-signed-url-for-update-response",
  "ResourceType" : "Custom::MyCustomResourceType",
  "LogicalResourceId" : "resource-logical-id",
  "PhysicalResourceId" : "provider-defined-physical-id",
  "ResourceProperties" : {
    "key1" : "new-string",
    "key2" : [ "new-list" ],
    "key3" : { "key4" : "new-map" }
  },
  "OldResourceProperties" : {
    "key1" : "string",
    "key2" : [ "list" ],
    "key3" : { "key4" : "map" }
  }
}
```

Delete

Cuando el desarrollador de la plantilla elimina la pila o elimina el recurso personalizado de la pila y, a continuación, actualiza la pila, CloudFormation envía una solicitud con RequestType establecido en Delete.

Elimine las solicitudes que contienen los siguientes campos:

RequestType

Delete.

Obligatorio: sí

Tipo: cadena

RequestId

Un ID único para la solicitud.

Obligatorio: sí

Tipo: cadena

StackId

El nombre de recurso de Amazon (ARN) que identifica la pila que contiene el recurso personalizado.

Obligatorio: sí

Tipo: cadena

ResponseURL

La URL de respuesta identifica un bucket de S3 prefirmado que recibe respuestas del proveedor de recursos personalizados para CloudFormation.

Obligatorio: sí

Tipo: cadena

ResourceType

El tipo de recurso elegido por el desarrollador de plantillas del recurso personalizado en la plantilla de CloudFormation. Los nombres de tipos de recursos personalizados pueden tener una longitud máxima de 60 caracteres y pueden incluir caracteres alfanuméricos y los siguientes caracteres: `_@-`.

Obligatorio: sí

Tipo: cadena

LogicalResourceId

El nombre elegido por el desarrollador de la plantilla (ID lógico) del recurso personalizado en la plantilla de CloudFormation.

Obligatorio: sí

Tipo: cadena

PhysicalResourceId

Un ID físico definido por el proveedor de recursos personalizados que es exclusivo de dicho proveedor.

Obligatorio: sí

Tipo: cadena

ResourceProperties

Este campo contiene el contenido del objeto `Properties` enviado por el desarrollador de la plantilla. El proveedor de recursos personalizado define su contenido.

Obligatorio: no

Tipo: objeto JSON

Ejemplo

```
{
  "RequestType" : "Delete",
  "RequestId" : "unique-request-id",
  "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/id",
  "ResponseURL" : "pre-signed-url-for-delete-response",
  "ResourceType" : "Custom::MyCustomResourceType",
  "LogicalResourceId" : "resource-logical-id",
  "PhysicalResourceId" : "provider-defined-physical-id",
  "ResourceProperties" : {
    "key1" : "string",
    "key2" : [ "list" ],
    "key3" : { "key4" : "map" }
  }
}
```

Objeto de respuesta

El proveedor de recursos personalizados envía una respuesta a la URL prefirmada para todos los tipos de recursos. Si el proveedor de recursos personalizados no envía una respuesta, CloudFormation espera hasta que se agote el tiempo de espera de la operación.

La respuesta debe ser un objeto JSON con los campos siguientes:

Status

Debe ser SUCCESS o FAILED.

Obligatorio: sí

Tipo: cadena

RequestId

Un ID único para la solicitud. Copie este valor exactamente como aparece en la solicitud.

Obligatorio: sí

Tipo: cadena

StackId

El nombre de recurso de Amazon (ARN) que identifica la pila que contiene el recurso personalizado. Copie este valor exactamente como aparece en la solicitud.

Obligatorio: sí

Tipo: cadena

LogicalResourceId

El nombre elegido por el desarrollador de la plantilla (ID lógico) del recurso personalizado en la plantilla de CloudFormation. Copie este valor exactamente como aparece en la solicitud.

Obligatorio: sí

Tipo: cadena

PhysicalResourceId

Este valor debe ser un identificador exclusivo del proveedor de recursos personalizados y puede tener un tamaño máximo de 1 KB. El valor debe ser una cadena que no esté vacía y debe ser idéntica para todas las respuestas del mismo recurso.

Al actualizar los recursos personalizados, el valor devuelto para `PhysicalResourceId` determina el comportamiento de la actualización. Si el valor devuelto es el mismo, CloudFormation lo considera una actualización normal. Si el valor cambia, CloudFormation interpreta la actualización como un reemplazo y envía una solicitud de eliminación al antiguo recurso. Para obtener más información, consulte [AWS::CloudFormation::CustomResource](#).

Obligatorio: sí

Tipo: cadena

Reason

Describe el motivo de una respuesta de error.

Obligatorio si `Status` es `FAILED`. De lo contrario, es opcional.

Obligatorio: condicional

Tipo: cadena

NoEcho

Indique si se enmascara la salida del recurso personalizado cuando se recupera con la función `Fn::GetAtt`. Si se establece en `true`, todos los valores devueltos se enmascaran con asteriscos (`*****`), excepto los almacenados en la sección *Metadata* de la plantilla. CloudFormation no transforma, modifica ni redacta ninguna información que incluya en la sección *Metadata*. El valor predeterminado es `false`.

Para obtener más información sobre el uso de `NoEcho` para enmascarar información confidencial, consulte la práctica recomendada de [No integre credenciales en sus plantillas](#).

Disponible solo para respuestas `Create` y `Update`. No se admite para las respuestas `Delete`.

Obligatorio: no

Tipo: booleano

Data

Los pares de nombre-valor definidos por el proveedor de recursos personalizados que se enviarán con la respuesta. Puede acceder a los valores proporcionados aquí por nombre en la plantilla con `Fn::GetAtt`.

Disponible solo para respuestas Create y Update. No se admite para las respuestas Delete.

 Important

Si los pares nombre-valor contienen información confidencial, debe usar el campo NoEcho para enmascarar la salida del recurso personalizado. De lo contrario, los valores son visibles a través de las API que muestran valores de propiedades (como DescribeStackEvents).

Obligatorio: no

Tipo: objeto JSON

Ejemplos de respuestas satisfactorias

Respuesta **Create** y **Update**

```
{
  "Status": "SUCCESS",
  "RequestId": "unique-request-id",
  "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/name/id",
  "LogicalResourceId": "resource-logical-id",
  "PhysicalResourceId": "provider-defined-physical-id",
  "NoEcho": true,
  "Data": {
    "key1": "value1",
    "key2": "value2"
  }
}
```

DeleteRespuesta de

```
{
  "Status": "SUCCESS",
  "RequestId": "unique-request-id",
  "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/name/id",
  "LogicalResourceId": "resource-logical-id",
  "PhysicalResourceId": "provider-defined-physical-id"
}
```

Respuesta de ejemplo con error

```
{
  "Status": "FAILED",
  "RequestId": "unique-request-id",
  "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/name/id",
  "LogicalResourceId": "resource-logical-id",
  "PhysicalResourceId": "provider-defined-physical-id",
  "Reason": "Required failure reason string"
}
```

Recursos personalizados y respaldados por Amazon SNS

En el siguiente tema se muestra cómo configurar un recurso personalizado con un token de servicio que especifica el tema de Amazon SNS al que CloudFormation envía las solicitudes. También aprenderá la secuencia de eventos y mensajes enviados y recibidos como resultado de la creación, las actualizaciones y la eliminación de pilas de recursos personalizados.

Con los recursos personalizados y Amazon SNS, puede habilitar escenarios como añadir nuevos recursos a una pila e inyectar datos dinámicos a una pila. Por ejemplo, al crear una pila, CloudFormation puede enviar una solicitud Create a un tema supervisado por una aplicación que se ejecute en una instancia de Amazon EC2. La notificación de Amazon SNS activa la aplicación para llevar a cabo tareas de aprovisionamiento adicionales, como recuperar un grupo de IP elásticas de direcciones IP. Después de este proceso, la aplicación envía una respuesta (y los datos de salida) que le notifica a CloudFormation que puede seguir con la operación de la pila.

Cuando se especifica un tema de Amazon SNS como destino de un recurso personalizado, CloudFormation envía mensajes al tema de SNS especificado durante las operaciones de pila que involucran el recurso personalizado. Para procesar estos mensajes y realizar las acciones necesarias, debe tener un punto de conexión compatible suscrito al tema de SNS.

Para obtener una introducción a los recursos personalizados y a su funcionamiento, consulte [Cómo funcionan los recursos personalizados](#). Para obtener más información sobre Amazon SNS y su funcionamiento, consulte la [Guía para desarrolladores de Amazon Simple Notification Service](#).

Uso de Amazon SNS para crear recursos personalizados

Temas

- [Paso 1: Creación de una pila](#)

- [Paso 2: Actualizaciones de pila](#)
- [Paso 3: Eliminación de pila](#)

Paso 1: Creación de una pila

1. El desarrollador de plantillas crea una pila de CloudFormation que contiene un recurso personalizado.

En el siguiente ejemplo de plantilla, usamos el nombre de tipo de recurso personalizado `Custom::SeleniumTester` para el recurso personalizado con ID lógico `MySeleniumTest`. Los nombres del tipo de recurso personalizado deben ser alfanuméricos y pueden tener una longitud máxima de 60 caracteres.

El tipo de recurso personalizado se declara con un token de servicio, propiedades específicas de proveedores opcionales y atributos [Fn::GetAtt](#) opcionales que define el proveedor de recursos personalizados. Se pueden usar estas propiedades y atributos para pasar a la información del template developer a custom resource provider y viceversa. El token de servicio especifica un tema de Amazon SNS que el proveedor de recursos ha configurado.

```
{
  "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

Los nombres y los valores de los datos a los que se accede con `Fn::GetAtt` los devuelve el proveedor de recursos personalizados durante la respuesta del proveedor a CloudFormation. Si custom resource provider es de terceros, template developer debe obtener los nombres de estos valores de devolución desde custom resource provider.

2. CloudFormation envía una notificación de Amazon SNS al proveedor de recursos con un `"RequestType" : "Create"` que contiene información sobre la pila, las propiedades de recursos personalizados de la plantilla de pila y una URL de S3 para la respuesta.

El tema de SNS que se usa para enviar la notificación está incrustado en la plantilla en la propiedad `ServiceToken`. Para evitar el uso de un valor codificado, un desarrollador de plantillas puede usar un parámetro de plantilla para que el valor se ingrese en el momento en que se lanza la pila.

El siguiente ejemplo muestra una solicitud de recurso `Create` personalizado que incluye un nombre de tipo de recurso personalizado, `Custom::SeleniumTester`, creado con un `LogicalResourceId` de `MySeleniumTester`:

```
{  
  "RequestType" : "Create",  
  "RequestId" : "unique-request-id",  
  "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-cd98-11ea-90d5-0a9cd3354c10",  
  "ResponseURL" : "http://pre-signed-S3-url-for-response",  
  "ResourceType" : "Custom::SeleniumTester",  
  "LogicalResourceId" : "MySeleniumTester",  
  "ResourceProperties" : {  
    "seleniumTester" : "SeleniumTest()",  
    "endpoints" : [ "http://mysite.com", "http://myecommercesite.com/", "http://search.mysite.com" ],  
    "frequencyOfTestsPerHour" : [ "3", "2", "4" ]  
  }  
}
```


Para obtener información detallada sobre el objeto de solicitud de las solicitudes `Create`, consulte el tema [Referencia de solicitud y respuesta](#).

3. El custom resource provider procesa los datos enviados por el template developer y determina si la solicitud `Create` se realizó correctamente. El proveedor de recursos usa luego la URL de S3 que CloudFormation ha enviado para remitir una respuesta `SUCCESS` o `FAILED`.

En función del tipo de respuesta, CloudFormation esperará diferentes campos de respuesta. Para obtener información sobre los campos de respuesta de un tipo de solicitud concreto, consulte la documentación correspondiente a ese tipo de solicitud en la sección [Referencia de solicitud y respuesta](#).

En respuesta a una solicitud de creación o actualización, el proveedor de recursos personalizados puede devolver los elementos de datos en el campo `Data` de la respuesta. Estos son pares de nombre/valor y los nombres corresponden a los atributos `Fn::GetAtt` que se utilizan con el recurso personalizado de la plantilla de pila. Los valores son los datos que se devuelven cuando el desarrollador de la plantilla llama a `Fn::GetAtt` por el recurso con el nombre de atributo.

El siguiente es un ejemplo de respuesta de recursos personalizados:

```
{
  "Status" : "SUCCESS",
  "RequestId" : "unique-request-id",
  "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-
cd98-11ea-90d5-0a9cd3354c10",
  "LogicalResourceId" : "MySeleniumTester",
  "PhysicalResourceId" : "Tester1",
  "Data" : {
    "resultsPage" : "http://www.myexampledomain/test-results/guid",
    "lastUpdate" : "2012-11-14T03:30Z"
  }
}
```

Para obtener información detallada sobre el objeto de respuesta de las solicitudes `Create`, consulte el tema [Referencia de solicitud y respuesta](#).

Los campos `StackId`, `RequestId` y `LogicalResourceId` deben copiarse textualmente de la solicitud.

- CloudFormation declara el estado de pila como `CREATE_COMPLETE` o `CREATE_FAILED`. Si la pila se creó correctamente, el desarrollador de plantillas puede utilizar los valores de salida del recurso personalizado que se creó, accediendo a ellos con [Fn::GetAtt](#).

Por ejemplo, la plantilla de recursos personalizados empleados para ilustrar usó `Fn::GetAtt` para copiar salidas de recursos en las salidas de pila:

```
"Outputs" : {
  "topItem" : {
    "Value" : { "Fn::GetAtt" : ["MySeleniumTest", "resultsPage"] }
  },
  "numRespondents" : {
    "Value" : { "Fn::GetAtt" : ["MySeleniumTest", "lastUpdate"] }
  }
}
```

Paso 2: Actualizaciones de pila

Para actualizar una pila existente, debe enviar una plantilla que especifique las actualizaciones de las propiedades de recursos de la pila, tal y como se muestra en el ejemplo siguiente. CloudFormation actualiza únicamente los recursos que tengan los cambios especificados en la plantilla. Para obtener más información, consulte [Comprensión de los comportamientos de actualización de los recursos de la pila](#).

Puede actualizar los recursos personalizados que requieren una sustitución del recurso físico subyacente. Al actualizar un recurso personalizado en una plantilla de CloudFormation, CloudFormation envía una solicitud de actualización a dicho recurso personalizado. Si un recurso personalizado requiere una sustitución, el nuevo recurso personalizado debe enviar una respuesta con el nuevo ID físico. Cuando CloudFormation recibe la respuesta, compara el `PhysicalResourceId` entre los recursos personalizados antiguos y nuevos. Si son diferentes, CloudFormation reconoce la actualización como sustitución y envía una solicitud de eliminación al antiguo recurso, tal y como se muestra en [Paso 3: Eliminación de pila](#).

Note

Si no realizó cambios en el recurso personalizado, CloudFormation no le enviará solicitudes durante la actualización de una pila.

1. template developer inicia una actualización de la pila que contiene un recurso personalizado. Durante una actualización, template developer puede especificar nuevas propiedades en la plantilla de pila.

El siguiente es un ejemplo de Update para la plantilla de pila que usa un tipo de recurso personalizado:

```
{
  "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 envía una notificación de Amazon SNS al proveedor de recursos con un "RequestType" : "Update" que contiene información similar a la llamada Create, salvo que el campo OldResourceProperties contiene las propiedades del antiguo recurso y ResourceProperties contiene las del recurso actualizado (si existe).

El siguiente es un ejemplo de una solicitud de Update:

```
{
  "RequestType" : "Update",
```

```

    "RequestId" : "unique-request-id",
    "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-
cd98-11ea-90d5-0a9cd3354c10",
    "ResponseURL" : "http://pre-signed-S3-url-for-response",
    "ResourceType" : "Custom::SeleniumTester",
    "LogicalResourceId" : "MySeleniumTester",
    "PhysicalResourceId" : "Tester1",
    "ResourceProperties" : {
        "seleniumTester" : "SeleniumTest()",
        "endpoints" : [ "http://mysite.com", "http://myecommercesite.com/", "http://
search.mysite.com",
        "http://mynewsite.com" ],
        "frequencyOfTestsPerHour" : [ "3", "2", "4", "3" ]
    },
    "OldResourceProperties" : {
        "seleniumTester" : "SeleniumTest()",
        "endpoints" : [ "http://mysite.com", "http://myecommercesite.com/", "http://
search.mysite.com" ],
        "frequencyOfTestsPerHour" : [ "3", "2", "4" ]
    }
}

```

Para obtener información detallada sobre el objeto de solicitud de las solicitudes Update, consulte el tema [Referencia de solicitud y respuesta](#).

3. El proveedor de recursos personalizados procesa los datos que CloudFormation envía. El recurso personalizado realiza la actualización y envía una respuesta SUCCESS o FAILED a la URL de S3. CloudFormation compara entonces los valores de PhysicalResourceIDs de los recursos personalizados antiguos y nuevos. Si son diferentes, CloudFormation reconoce que la actualización requiere una sustitución y envía una solicitud de eliminación al antiguo recurso. El siguiente ejemplo demuestra la respuesta de custom resource provider a una solicitud de Update.

```

{
    "Status" : "SUCCESS",
    "RequestId" : "unique-request-id",
    "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-
cd98-11ea-90d5-0a9cd3354c10",
    "LogicalResourceId" : "MySeleniumTester",
    "PhysicalResourceId" : "Tester2"
}

```

Para obtener información detallada sobre el objeto de respuesta de las solicitudes Update, consulte el tema [Referencia de solicitud y respuesta](#).

Los campos `StackId`, `RequestId` y `LogicalResourceId` deben copiarse textualmente de la solicitud.

4. CloudFormation declara el estado de pila como `UPDATE_COMPLETE` o `UPDATE_FAILED`. Si falla la actualización, la pila vuelve a su estado anterior. Si la pila se ha actualizado correctamente, `template developer` puede obtener acceso a cualquier nuevo valor de salida del recurso personalizado creado con `Fn::GetAtt`.

Paso 3: Eliminación de pila

1. El desarrollador de plantillas elimina una pila que contiene un recurso personalizado. CloudFormation obtiene las propiedades actuales especificadas en la plantilla de pila junto con el tema de SNS y se prepara para realizar una solicitud al proveedor de recursos personalizados.
2. CloudFormation envía una notificación de Amazon SNS al proveedor de recursos con un `"RequestType" : "Delete"` que contiene información actual sobre la pila, las propiedades de recursos personalizados de la plantilla de pila y una URL de S3 para la respuesta.

Cuando se elimina una pila o se hace una actualización que elimina o sustituye el recurso personalizado, CloudFormation compara el valor de `PhysicalResourceId` de los recursos personalizados antiguos con los nuevos. Si son diferentes, CloudFormation reconoce la actualización como sustitución y envía una solicitud de eliminación para el recurso anterior (`OldPhysicalResource`), tal y como se muestra en el siguiente ejemplo de una solicitud `Delete`.

```
{
  "RequestType" : "Delete",
  "RequestId" : "unique-request-id",
  "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-cd98-11ea-90d5-0a9cd3354c10",
  "ResponseURL" : "http://pre-signed-S3-url-for-response",
  "ResourceType" : "Custom::SeleniumTester",
  "LogicalResourceId" : "MySeleniumTester",
  "PhysicalResourceId" : "Tester1",
  "ResourceProperties" : {
    "seleniumTester" : "SeleniumTest()",
  }
}
```

```
"endpoints" : [ "http://mysite.com", "http://myecommercesite.com/", "http://search.mysite.com",  
               "http://mynewsite.com" ],  
"frequencyOfTestsPerHour" : [ "3", "2", "4", "3" ]  
}
```

Para obtener información detallada sobre el objeto de solicitud de las solicitudes Delete, consulte el tema [Referencia de solicitud y respuesta](#).

DescribeStackResource, DescribeStackResources y ListStackResources muestran el nombre definido por el usuario en caso de que se haya especificado.

3. El proveedor de recursos personalizados procesa los datos enviados por CloudFormation y determina si la solicitud Delete se realizó correctamente. El proveedor de recursos usa luego la URL de S3 que CloudFormation ha enviado para remitir una respuesta SUCCESS o FAILED. Para poder eliminar correctamente una pila con un recurso personalizado, el custom resource provider debe responder correctamente a una solicitud de eliminación.

A continuación se muestra un ejemplo de respuesta de custom resource provider a una solicitud de Delete:

```
{  
  "Status" : "SUCCESS",  
  "RequestId" : "unique-request-id",  
  "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-cd98-11ea-90d5-0a9cd3354c10",  
  "LogicalResourceId" : "MySeleniumTester",  
  "PhysicalResourceId" : "Tester1"  
}
```

Para obtener información detallada sobre el objeto de respuesta de las solicitudes Delete, consulte el tema [Referencia de solicitud y respuesta](#).

Los campos StackId, RequestId y LogicalResourceId deben copiarse textualmente de la solicitud.

4. CloudFormation declara el estado de pila como DELETE_COMPLETE o DELETE_FAILED.

Recursos personalizados y respaldados por Lambda

Al asociar una función de Lambda con un recurso personalizado, la función se invoca siempre que se crea, actualiza o elimina el recurso personalizado. CloudFormation llama a una API de Lambda para invocar la función y transferir todos los datos de la solicitud (como el tipo de solicitud y las propiedades de recursos) a la función. La potencia y la posibilidad de personalización de funciones de Lambda en combinación con CloudFormation permiten una amplia variedad de situaciones, como, por ejemplo, las búsquedas de ID de AMI de forma dinámica durante la creación de la pila o la implementación y el uso de las funciones de utilidad, tales como las funciones de reversión de cadenas.

Para obtener una introducción a los recursos personalizados y a su funcionamiento, consulte [Cómo funcionan los recursos personalizados](#).

Temas

- [Explicación: cómo crear un mecanismo de retardo mediante un recurso personalizado respaldado por Lambda](#)
- [cfn-responseMódulo de](#)

Explicación: cómo crear un mecanismo de retardo mediante un recurso personalizado respaldado por Lambda

En este tutorial, se muestra cómo configurar e inicializar un recurso personalizado respaldado por Lambda mediante una plantilla de muestra de CloudFormation. Esta plantilla crea un mecanismo de retraso que pausa las implementaciones de pilas durante un tiempo específico. Esto puede resultar útil cuando se necesita introducir retrasos deliberados durante el aprovisionamiento de recursos; por ejemplo, cuando se espera a que los recursos se establezcan antes de que se creen los recursos dependientes.

Note

Si bien anteriormente se recomendaban recursos personalizados respaldados por Lambda para recuperar los ID de AMI, ahora recomendamos usar parámetros AWS Systems Manager. Este enfoque hace que sus plantillas sean más reutilizables y fáciles de mantener. Para obtener más información, consulte [Obtención de un valor de texto sin formato del Almacén de parámetros de Systems Manager](#).

Temas

- [Descripción general](#)
- [Plantilla de muestra](#)
- [Explicación de la plantilla de ejemplo](#)
- [Requisitos previos](#)
- [Lanzamiento de la pila](#)
- [Saneamiento de recursos](#)
- [Información relacionada](#)

Descripción general

La plantilla de pila de ejemplo utilizada en este tutorial crea un recurso personalizado respaldado por Lambda. Este recurso personalizado introduce un retraso configurable (60 segundos por defecto) durante la creación de la pila. El retraso se produce durante las actualizaciones de la pila solo cuando se modifican las propiedades del recurso personalizado.

Esta plantilla proporciona los siguientes recursos:

- un recurso personalizado,
- una función de Lambda y
- un rol de IAM que permite a Lambda escribir registros en CloudWatch.

También define dos salidas:

- El tiempo real que esperó la función.
- Un identificador único generado durante cada ejecución de la función de Lambda.

Note

CloudFormation es un servicio gratuito, pero Lambda cobra en función del número de solicitudes de sus funciones y del tiempo de ejecución de su código. Para obtener más información acerca de los precios de Lambda, consulte [Precios de AWS Lambda](#).

Plantilla de muestra

A continuación, puede ver la plantilla de ejemplo de un recurso personalizado respaldado por Lambda con el mecanismo de retardo:

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\"][\"ServiceTimeout\
\", 0))",
        "        sleep(wait_seconds)",
        "        response = {",
        "            \"TimeWaited\": wait_seconds,",
        "            \"Id\": id ",
        "        }",
        "        cfnresponse.send(event, context, cfnresponse.SUCCESS, response,
        \"Waiter-\"+id)
    ]}]
}
},
"CFNWaiterCustomResource": {
    "Type": "AWS::CloudFormation::CustomResource",
    "Properties": {
        "ServiceToken": { "Fn::GetAtt": ["CFNWaiter", "Arn"] },
        "ServiceTimeout": 60
    }
}
},
"Outputs": {
    "TimeWaited": {
        "Value": { "Fn::GetAtt": ["CFNWaiterCustomResource", "TimeWaited"] },
        "Export": { "Name": "TimeWaited" }
    },
    "WaiterId": {
        "Value": { "Fn::GetAtt": ["CFNWaiterCustomResource", "Id"] },
        "Export": { "Name": "WaiterId" }
    }
}
}
}

```

YAML

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

```
Resources:
  LambdaExecutionRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Statement:
          - Effect: "Allow"
            Principal:
              Service:
                - "lambda.amazonaws.com"
            Action:
              - "sts:AssumeRole"
      Path: "/"
      Policies:
        - PolicyName: "AllowLogs"
          PolicyDocument:
            Statement:
              - Effect: "Allow"
                Action:
                  - "logs:*"
                Resource: "*"
  CFNWaiter:
    Type: AWS::Lambda::Function
    Properties:
      Handler: index.handler
      Runtime: python3.9
      Timeout: 900
      Role: !GetAtt LambdaExecutionRole.Arn
      Code:
        ZipFile:
          !Sub |
            from time import sleep
            import json
            import cfnresponse
            import uuid

            def handler(event, context):
                wait_seconds = 0
                id = str(uuid.uuid1())
                if event["RequestType"] in ["Create", "Update"]:
                    wait_seconds = int(event["ResourceProperties"].get("ServiceTimeout", 0))
                    sleep(wait_seconds)
                response = {
                    "TimeWaited": wait_seconds,
```

```

        "Id": id
    }
    cfnresponse.send(event, context, cfnresponse.SUCCESS, response,
    "Waiter-"+id)
    CFNWaiterCustomResource:
      Type: AWS::CloudFormation::CustomResource
      Properties:
        ServiceToken: !GetAtt CFNWaiter.Arn
        ServiceTimeout: 60
    Outputs:
      TimeWaited:
        Value: !GetAtt CFNWaiterCustomResource.TimeWaited
        Export:
          Name: TimeWaited
      WaiterId:
        Value: !GetAtt CFNWaiterCustomResource.Id
        Export:
          Name: WaiterId

```

Explicación de la plantilla de ejemplo

Los siguientes fragmentos explican partes relevantes de la plantilla de ejemplo que ayudan a comprender cómo la función de Lambda se asocia a un recurso personalizado y a entender la salida.

CFNWaiter del recurso [AWS::Lambda::Function](#)

El recurso `AWS::Lambda::Function` especifica el código fuente de la función, el nombre del controlador, el entorno del tiempo de ejecución y el nombre de recurso de Amazon (ARN).

La propiedad `Handler` está establecida en `index.handler`, ya que utiliza un código fuente de Python. Para obtener más información sobre los identificadores de controlador aceptados al usar código fuente de funciones en línea, consulte [AWS::Lambda::Function Code](#).

El `Runtime` se especifica como `python3.9`, ya que el archivo de origen está escrito en Python.

El `Timeout` se establece en 900 segundos.

La propiedad `Role` utiliza la función `Fn::GetAtt` para obtener el ARN del rol de ejecución `LambdaExecutionRole` que se declara en el recurso `AWS::IAM::Role` en la plantilla.

La propiedad `Code` define el código de la función en línea mediante una función de Python. La función de Python en la plantilla de ejemplo hace lo siguiente:

- Crear un ID único utilizando el UUID
- Verifica si se trata de una solicitud de creación o de actualización
- Duerme durante el tiempo especificado por `ServiceTimeout` durante solicitudes de `Create` o `Update`
- Devuelve el tiempo de espera y el identificador único

JSON

```
...
    "CFNWaiter": {
        "Type": "AWS::Lambda::Function",
        "Properties": {
            "Handler": "index.handler",
            "Runtime": "python3.9",
            "Timeout": 900,
            "Role": { "Fn::GetAtt": ["LambdaExecutionRole", "Arn"] },
            "Code": {
                "ZipFile": { "Fn::Join": ["\n", [
                    "from time import sleep",
                    "import json",
                    "import cfnresponse",
                    "import uuid",
                    "",
                    "def handler(event, context):",
                    "    wait_seconds = 0",
                    "    id = str(uuid.uuid1())",
                    "    if event[\"RequestType\"] in [\"Create\", \"Update\"]:",
                    "        wait_seconds = int(event[\"ResourceProperties\"].get(\"ServiceTimeout", 0))",
                    "        sleep(wait_seconds)",
                    "        response = {",
                    "            \"TimeWaited\": wait_seconds,",
                    "            \"Id\": id ",
                    "        }",
                    "        cfnresponse.send(event, context, cfnresponse.SUCCESS, response,",
                    "\"Waiter-\"+id)",
                    ]]]
                }
            }
        },
        ...
```

YAML

```
...
CFNWaiter:
  Type: AWS::Lambda::Function
  Properties:
    Handler: index.handler
    Runtime: python3.9
    Timeout: 900
    Role: !GetAtt LambdaExecutionRole.Arn
    Code:
      ZipFile:
        !Sub |
        from time import sleep
        import json
        import cfnresponse
        import uuid

        def handler(event, context):
            wait_seconds = 0
            id = str(uuid.uuid1())
            if event["RequestType"] in ["Create", "Update"]:
                wait_seconds = int(event["ResourceProperties"].get("ServiceTimeout", 0))
                sleep(wait_seconds)
            response = {
                "TimeWaited": wait_seconds,
                "Id": id
            }
            cfnresponse.send(event, context, cfnresponse.SUCCESS, response,
"Waiter-"+id)
...

```

LambdaExecutionRole del recurso [AWS::IAM::Role](#)

El recurso `AWS::IAM::Role` crea un rol de ejecución para la función de Lambda, con una política de asunción que autoriza a Lambda a utilizar dicho rol. También contiene una política que permite el acceso a Registros de CloudWatch.

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: "*"
...

```

CFNWaiterCustomResource de recurso [AWS::CloudFormation::CustomResource](#)

El recurso personalizado se vincula a la función de Lambda con su ARN mediante `!GetAtt CFNWaiter.Arn`. Establecerá un tiempo de espera de 60 segundos para las operaciones de creación y actualización, conforme a lo dispuesto en `ServiceTimeout`. El recurso solo se invocará para una operación de actualización si se modifican las propiedades.

JSON

```

...
  "CFNWaiterCustomResource": {
    "Type": "AWS::CloudFormation::CustomResource",
    "Properties": {
      "ServiceToken": { "Fn::GetAtt": ["CFNWaiter", "Arn"] },
      "ServiceTimeout": 60
    }
  },
...

```

YAML

```

...
CFNWaiterCustomResource:
  Type: AWS::CloudFormation::CustomResource
  Properties:
    ServiceToken: !GetAtt CFNWaiter.Arn
    ServiceTimeout: 60
...

```

Outputs

Esta plantilla produce como Outputs el `TimeWaited` y el `WaiterId`. El valor `TimeWaited` utiliza una función de `Fn::GetAtt` para indicar la cantidad de tiempo que el recurso de espera

realmente esperó. El `WaiterId` utiliza una función de `Fn::GetAtt` para proporcionar el identificador único que se generó y se asoció a la ejecución.

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
...
```

Requisitos previos

Se necesitan permisos de IAM para utilizar todos los servicios correspondientes, como Lambda y CloudFormation.

Lanzamiento de la pila

Para crear la pila

1. Busque la plantilla que prefiera (YAML o JSON) en la sección [Plantilla de muestra](#) y guárdela en la máquina con el nombre `samplelambdabackedcustomresource.template`.
2. Abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation/>.
3. En la página Pilas, seleccione Crear pila en la parte superior derecha y, a continuación, seleccione Con recursos nuevos (estándar).
4. En Requisito previo: preparar la plantilla, elija Seleccione una plantilla existente.
5. En Especificar plantilla, seleccione Cargar un archivo de plantilla y, a continuación, elija Elegir archivo.
6. Seleccione el archivo de plantilla `samplelambdabackedcustomresource.template` que guardó anteriormente.
7. Elija Siguiente.
8. En Nombre de la pila, escriba **SampleCustomResourceStack** y, a continuación, elija Siguiente.
9. Para este tutorial, no tiene que añadir etiquetas ni especificar una configuración avanzada, así que elija Next (Siguiente).
10. Verifique que el nombre de la pila sea correcto y luego seleccione Crear.

CloudFormation puede tardar varios minutos en crear la pila. Para monitorizar el progreso, vea los eventos de la pila. Para obtener más información, consulte [Visualización de la información de la pila desde la consola de CloudFormation](#).

Si se crea la pila correctamente, se crean todos los recursos de la pila, como la función Lambda y el recurso personalizado. Ha utilizado correctamente una función de Lambda y un recurso personalizado.

Si la función de Lambda devuelve un error, vea los registros de la función en la [consola](#) de Registros de CloudWatch. El nombre de la flujo de registros es el ID físico del recurso personalizado, que puede encontrar al visualizar los recursos de la pila. Para obtener más información, consulte la sección de [Visualización de datos de registro](#) en la Guía del usuario de Amazon CloudWatch.

Saneamiento de recursos

Elimine la pila para limpiar todos los recursos de la pila que creó para que no se le cobren los recursos innecesarios.

Para eliminar la pila

1. En la consola de CloudFormation, elija la pila SampleCustomResourceStack.
2. Elija Actions (Acciones) y, a continuación, Delete Stack (Eliminar pila).
3. En el mensaje de confirmación, elija Yes, Delete (Sí, eliminar).

Se eliminan todos los recursos que creó.

Ahora que comprende cómo crear y utilizar un recurso personalizado respaldado por Lambda, puede utilizar la plantilla y el código de ejemplo de esta guía para crear y experimentar con otras pilas y funciones.

Información relacionada

- [Referencia de recursos personalizados de CloudFormation](#)
- [AWS::CloudFormation::CustomResource](#)

cfn-responseMódulo de

En su plantilla de CloudFormation puede especificar una función de Lambda como destino de un recurso personalizado. Al utilizar la propiedad `ZipFile` para especificar el código fuente de la [función](#), puede cargar el módulo `cfn-response` para enviar las respuestas desde su función de Lambda a un recurso personalizado. El módulo `cfn-response` es una biblioteca que simplifica el envío de respuestas al recurso personalizado que invocó la función de Lambda. El módulo tiene un método `send` que envía un [objeto de respuesta](#) a un recurso personalizado mediante una URL prefirmada de Amazon S3 (`ResponseURL`).

El módulo `cfn-response` solo está disponible si utiliza la propiedad `ZipFile` para escribir el código fuente. No está disponible para el código fuente que se almacena en buckets de Amazon S3. Para un código de buckets de , debe escribir sus propias funciones para enviar respuestas.

 Note

Después de ejecutar el método `send`, la función de Lambda termina, así que todo lo que escriba después de este método se pasa por alto.

Carga del módulo **cfn-response**

Para las funciones de Node.js, utilice la función `require()` para cargar el módulo `cfn-response`. Por ejemplo, el siguiente ejemplo de código crea un objeto `cfn-response` con el nombre `response`:

```
var response = require('cfn-response');
```

Para Python, utilice la declaración `import` para cargar el módulo `cfnresponse`, tal y como se muestra en el ejemplo siguiente:

 Note

Utilice esta declaración de importación exacta. Si utiliza otras variantes de la instrucción de importación, CloudFormation no incluirá el módulo de respuesta.

```
import cfnresponse
```

sendParámetros del método

Puede utilizar los siguientes parámetros con el método `send`.

`event`

Los campos de una [solicitud de recurso personalizado](#).

`context`

Un objeto, específico de las funciones de Lambda, que puede utilizar para especificar cuando la función y cualquier devolución de llamada han completado la ejecución o para obtener acceso a información desde el entorno de ejecución de Lambda. Para obtener más información, consulte [Creación de funciones de Lambda con Node.js](#) en la Guía para desarrolladores de AWS Lambda.

responseStatus

Independientemente de que la función se complete con éxito. Utilice las constantes del módulo `cfnresponse` para especificar el estado: `SUCCESS` para las ejecuciones de éxito y `FAILED` para ejecuciones fallidas.

responseData

El campo `Data` de un [objeto de respuesta](#) de recursos personalizados. Los datos consisten en una lista de pares de nombre y valor.

physicalResourceId

Opcional. El identificador único del recurso personalizado que invocó la función. De forma predeterminada, el módulo utiliza el nombre de la secuencia de registros de Amazon CloudWatch Logs que se asocia a la función de Lambda.

El valor devuelto para un `PhysicalResourceId` puede cambiar las operaciones de actualización de recursos personalizadas. Si el valor devuelto es el mismo, se considera una actualización normal. Si el valor devuelto es diferente, CloudFormation reconoce la actualización como un reemplazo y envía una solicitud de eliminación al antiguo recurso. Para obtener más información, consulte [AWS::CloudFormation::CustomResource](#).

noEcho

Opcional. Indique si se enmascara la salida del recurso personalizado cuando se recupera con la función `Fn::GetAtt`. Si se establece en `true`, todos los valores devueltos se enmascaran con asteriscos (`*****`), excepto la información almacenada en las ubicaciones especificadas a continuación. De forma predeterminada, este valor es `false`.

Important

El uso del atributo `NoEcho` no enmascara ninguna información almacenada en lo que se muestra a continuación:

- La sección de la plantilla `Metadata`. CloudFormation no transforma, modifica ni redacta ninguna información que incluya en la sección `Metadata`. Para obtener más información, consulte [Metadata](#).
- La sección de la plantilla `Outputs`. Para obtener más información, consulte [Salidas](#).
- El atributo `Metadata` de una definición de recurso. Para obtener más información, consulte [Metadata atributo](#).

Recomendamos encarecidamente que no utilice estos mecanismos para incluir información confidencial, como contraseñas o secretos.

Para obtener más información sobre el uso de NoEcho para enmascarar información confidencial, consulte la práctica recomendada de [No integre credenciales en sus plantillas](#).

Ejemplos

Node.js

En el siguiente ejemplo Node.js, la función de Lambda en línea toma un valor de entrada y lo multiplica por 5. Las funciones en línea son especialmente útiles para las funciones más pequeñas, ya que le permiten especificar el código fuente directamente en la plantilla, en lugar de crear un paquete y cargarlo en un bucket de Amazon S3. La función utiliza el método `cfn-response send` para enviar el resultado de vuelta al recurso personalizado que invocó.

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

En el siguiente ejemplo de Python, la función de Lambda en línea toma un valor de entero y lo multiplica por 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")
```

Código fuente del módulo

Temas

- [Código fuente de Node.js asíncrono](#)
- [Código fuente de Node.js](#)
- [Código fuente Python](#)

Código fuente de Node.js asíncrono

A continuación, se muestra el código fuente del módulo de respuesta para las funciones de Node.js si el controlador es asíncrono. Revíselo para entender qué hace el módulo y para obtener ayuda con la implementación de sus propias funciones de respuesta.

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: MIT-0

exports.SUCCESS = "SUCCESS";
exports.FAILED = "FAILED";

exports.send = function(event, context, responseStatus, responseData,
    physicalResourceId, noEcho) {

    return new Promise((resolve, reject) => {
        var responseBody = JSON.stringify({
            Status: responseStatus,
            Reason: "See the details in CloudWatch Log Stream: " +
context.logStreamName,
            PhysicalResourceId: physicalResourceId || context.logStreamName,
            StackId: event.StackId,
            RequestId: event.RequestId,
            LogicalResourceId: event.LogicalResourceId,
            NoEcho: noEcho || false,
            Data: responseData
        });

        console.log("Response body:\n", responseBody);

        var https = require("https");
        var url = require("url");

        var parsedUrl = url.parse(event.ResponseURL);
        var options = {
            hostname: parsedUrl.hostname,
            port: 443,
            path: parsedUrl.path,
            method: "PUT",
            headers: {
                "content-type": "",
                "content-length": responseBody.length
            }
        };

        var request = https.request(options, function(response) {
            console.log("statusCode: ", response.statusCode);
        });

        request.write(responseBody);
        request.end();

        return request;
    });
};
```



```

    };

    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=*****');
}

```

Código fuente de Node.js

A continuación, se muestra el código fuente del módulo de respuesta para las funciones de Node.js si el controlador no es asíncrono. Revíselo para entender qué hace el módulo y para obtener ayuda con la implementación de sus propias funciones de respuesta.

```

// 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();
}
```

Código fuente Python

A continuación, se muestra el código fuente del módulo de respuesta de las funciones de 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)
```

Procesamientos personalizados en plantillas de CloudFormation con macros de plantillas

Las macros le permiten realizar un procesamiento personalizado de plantillas, desde acciones sencillas como operaciones de búsqueda y reemplazo, hasta amplias transformaciones de plantillas completas.

Para que se haga una idea de la gran variedad de posibilidades, piense en las transformaciones `AWS::Include` y `AWS::Serverless`, que son macros alojadas por CloudFormation:

- La [transformación `AWS::Include`](#) le permite insertar fragmentos de código reutilizable en las plantillas.
- La [transformación `AWS::Serverless`](#) toma una plantilla completa escrita en la sintaxis de AWS Serverless Application Model (AWS SAM) y la transforma y amplía en una plantilla de CloudFormation compatible. Para obtener más información sobre las aplicaciones sin servidor y AWS SAM, consulte la [Guía para desarrolladores de AWS Serverless Application Model](#).

Temas

- [Facturación](#)
- [Ejemplos de macros](#)
- [Recursos relacionados](#)
- [Descripción general de macros de CloudFormation](#)
- [Creación de una definición de macro de CloudFormation](#)
- [Ejemplo de macro sencilla de reemplazo de cadenas](#)
- [Solución de problemas de la plantilla procesada](#)

Facturación

Cuando se ejecuta una macro, se factura al propietario de la función de Lambda todos los gastos relacionados con la ejecución de esa función.

Las transformaciones `AWS::Include` y `AWS::Serverless` son macros alojadas por CloudFormation. No se cobra por usarlas.

Ejemplos de macros

Además de los ejemplos de esta sección, encontrará ejemplos de macros, incluido el código fuente y las plantillas, en nuestro [repositorio de GitHub](#). Estos ejemplos se proporcionan “tal cual” para fines pedagógicos.

Recursos relacionados

- [AWS::CloudFormation::Macro](#)
- [Sección Transform de la plantilla de CloudFormation](#)
- [Fn::Transform](#)
- [AWS::Serverless Transformación de](#)
- [AWS::Include Transformación de](#)

Descripción general de macros de CloudFormation

El procesamiento de plantillas con macros consta de dos fases principales: crear la macro en sí y luego usar la macro para procesar las plantillas.

Para crear una definición de macro, debe crear lo siguiente:

- Una función de Lambda para realizar el procesamiento de la plantilla. Esta función de Lambda acepta un fragmento de código o bien la totalidad de la plantilla, así como los parámetros adicionales que se definan. Devuelve el fragmento de código de plantilla procesados o toda la plantilla como respuesta.
- Un recurso del tipo [AWS::CloudFormation::Macro](#), para que los usuarios puedan llamar a la función de Lambda desde plantillas de CloudFormation. Este recurso especifica el ARN de la función de Lambda para invocar esta macro y propiedades opcionales adicionales para ayudar con la depuración. Para crear este recurso dentro de una cuenta, cree una plantilla de pila que

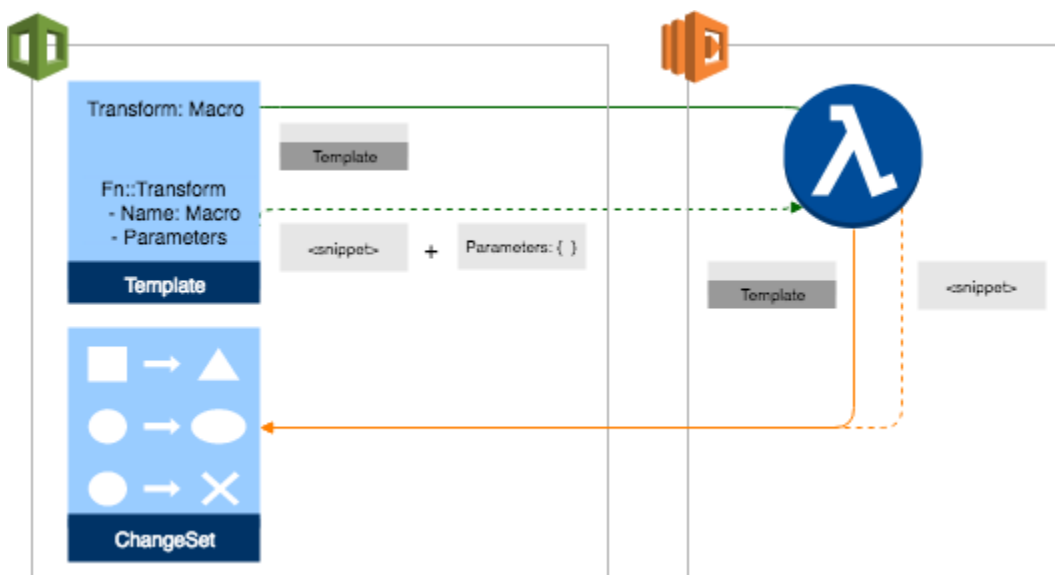
incluya el recurso `AWS::CloudFormation::Macro` y, a continuación, cree otra pila a partir de la plantilla. CloudFormation StackSets actualmente no es compatible con la creación o la actualización de conjuntos de pilas con permisos administrados por servicios a partir de plantillas que hacen referencia a macros de .

Para utilizar una macro, consulte la macro en su plantilla:

- Para procesar una sección o un fragmento de una plantilla, consulte la macro en una función `Fn::Transform` que se encuentre ubicada en relación con el contenido de la plantilla que desea transformar. Cuando utilice `Fn::Transform`, también puede transferir todos los parámetros especificados que requiera.
- Para procesar una plantilla completa, consulte la macro en la sección [Transform](#) de la plantilla.

A continuación, se suele crear un conjunto de cambios y, a continuación, ejecutarlo. (El procesamiento de macros puede agregar varios recursos de los que quizás no esté al tanto. Para asegurarse de que está al tanto de todos los cambios hechos por las macros, le recomendamos usar conjuntos de cambios). CloudFormation pasa el contenido de la plantilla especificada, junto con cualquier parámetro especificado adicional, a la función de Lambda especificada en el recurso de macro. La función de Lambda devuelve el contenido de la plantilla procesada, ya sea como fragmento o como una plantilla completa.

Después de que se hayan llamado a todas las macros de la plantilla, CloudFormation genera un conjunto de cambios que incluye el contenido de la plantilla procesada. Después de revisar el conjunto de cambios, ejecútelo para aplicar los cambios.



Cómo crear pilas directamente

Para crear o actualizar una pila utilizando una plantilla que hace referencia a macros, normalmente debe crear un conjunto de cambios y ejecutarlo. Un conjunto de cambios describe las acciones que CloudFormation realizará en función de la plantilla procesada. El procesamiento de macros puede añadir numerosos recursos que usted podría desconocer. Para asegurarse de estar al tanto de todos los cambios hechos por las macros, sugerimos usar conjuntos de cambios. Después de revisar el conjunto de cambios, puede ejecutarlo para aplicar realmente los cambios.

Una macro puede añadir recursos de IAM a su plantilla. Para estos recursos, CloudFormation necesita que [reconozca sus capacidades](#). Dado que CloudFormation no puede saber qué recursos se agregan antes de procesar su plantilla, es posible que tenga que reconocer las capacidades de IAM al crear el conjunto de cambios, en función de si las macros a las que se hace referencia contienen recursos de IAM. De este modo, al ejecutar el conjunto de cambios, CloudFormation tiene las capacidades necesarias para crear recursos de IAM.

Para crear o actualizar una pila directamente desde una plantilla procesada, sin antes haber revisado los cambios propuestos en un conjunto de cambios, especifique la capacidad `CAPABILITY_AUTO_EXPAND` durante una solicitud `CreateStack` o `UpdateStack`. Solo debe crear pilas directamente desde una plantilla de pila que contiene macros si sabe qué procesamiento realiza la macro. No puede utilizar conjuntos de cambios con macros de conjuntos de pilas y debe actualizar el conjunto de pilas directamente.

Para obtener más información consulte [CreateStack](#) o [UpdateStack](#) en la Referencia de la API de AWS CloudFormation.

Important

Si la plantilla del conjunto de pilas hace referencia a una o más macros, debe crear el conjunto de pilas directamente desde la plantilla procesada, sin revisar primero los cambios resultantes en un conjunto de cambios. El procesamiento de macros puede añadir numerosos recursos que usted podría desconocer. Antes de crear o actualizar un conjunto de pilas a partir de una plantilla que haga referencia directamente a macros, asegúrese de saber qué procesamiento realiza cada macro.

Para reducir la cantidad de pasos para lanzar pilas desde plantillas que hagan referencia a macros, puede usar los comandos `package` y `deploy` de la AWS CLI. Para obtener más información,

consulte [Carga de artefactos locales en un bucket de S3 con la AWS CLI](#) y [Crear una pila que incluya transformaciones](#).

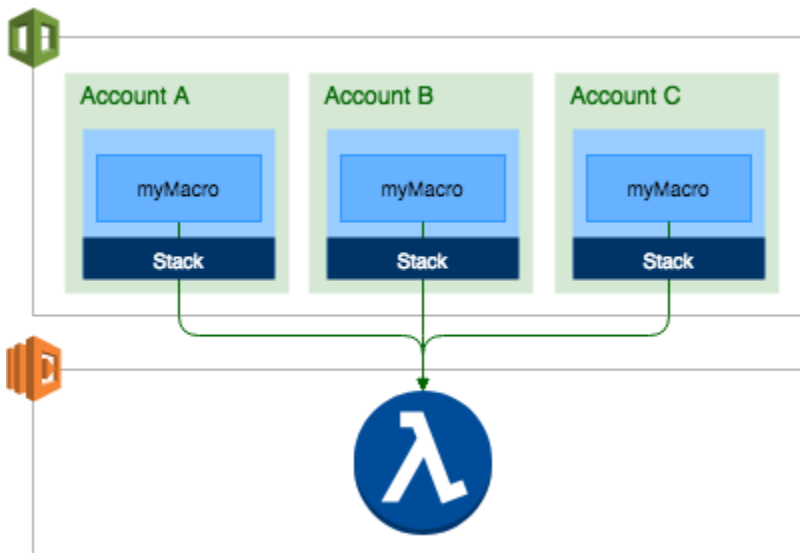
Consideraciones

Cuando trabaje con macros, tenga en cuenta las siguientes notas y limitaciones:

- Las macros se admiten únicamente en Regiones de AWS donde Lambda está disponible. Para ver una lista de las regiones en las que Lambda está disponible, consulte [Puntos de conexión y cuotas de AWS Lambda](#).
- Todos los fragmentos de código de plantilla procesados tienen que ser un JSON válido.
- Cualquier fragmento de plantilla procesado debe pasar las comprobaciones de validación para una operación de creación de pila, actualización de pila, creación de conjunto de pilas o actualización de conjunto de pilas.
- CloudFormation resuelve en primer lugar las macros y, a continuación, procesa la plantilla. La plantilla obtenida tiene que ser un JSON válido y no debe superar el límite de tamaño de la plantilla.
- Debido al orden en que CloudFormation procesa los elementos de una plantilla, una macro no puede incluir módulos en el contenido de la plantilla procesada que devuelve a CloudFormation. Para obtener más información, consulte [Orden de evaluación de macro](#).
- Cuando se utiliza la característica de restauración de actualizaciones, CloudFormation utiliza una copia de la plantilla original. Se restaura en la plantilla original incluso si el fragmento de código incluido se ha cambiado.
- La inclusión de macros dentro de otras macros no funciona porque no procesamos las macros de forma recursiva.
- La función intrínseca `Fn::ImportValue` no es compatible actualmente en las macros.
- Las funciones intrínsecas incluidas en la plantilla se evalúan después de cualquier macro. Por lo tanto, el contenido de plantilla procesado que su macro devuelve puede incluir llamadas a funciones intrínsecas y se evalúan como de costumbre.
- Actualmente, StackSets no es compatible con la creación o la actualización de conjuntos de pilas con permisos administrados por servicios a partir de plantillas que hacen referencia a macros de CloudFormation.

Alcance y permisos de la cuenta de macro

Puede utilizar macros solo en la cuenta en la que se han creado como un recurso. El nombre de la macro tiene que ser único en una cuenta determinada. Sin embargo, puede hacer que la misma funcionalidad esté disponible en varias cuentas habilitando el acceso entre cuentas en la función de Lambda subyacente y, a continuación, creando definiciones de macro que haga referencia a dicha función en varias cuentas. En el ejemplo siguiente, tres cuentas macro contienen definiciones y cada una de ellas apunta a la misma función de Lambda.



Para crear una definición de macro, el usuario debe tener permisos para crear una pila o un conjunto de pilas dentro de la cuenta especificada.

Para que CloudFormation ejecute correctamente una macro incluida en una plantilla, el usuario debe tener permisos Invoke para la función de Lambda subyacente. Para evitar posibles transmisiones de permisos, CloudFormation representa al usuario al ejecutar la macro.

Para obtener más información, consulte [Administrar permisos en AWS Lambda](#) en la Guía para desarrolladores de AWS Lambda y [Acciones, recursos y claves de condición para AWS Lambda](#) en la Referencia de autorizaciones de servicio.

Creación de una definición de macro de CloudFormation

Al crear una definición de macro, la definición de macro hace que la función de Lambda subyacente esté disponible en la cuenta especificada, de modo que CloudFormation la llame para procesar las plantillas.

Mapeo de eventos

Cuando CloudFormation invoca la función de Lambda de una macro, envía una solicitud en formato JSON con la siguiente estructura:

```
{
  "region" : "us-east-1",
  "accountId" : "$ACCOUNT_ID",
  "fragment" : { ... },
  "transformId" : "$TRANSFORM_ID",
  "params" : { ... },
  "requestId" : "$REQUEST_ID",
  "templateParameterValues" : { ... }
}
```

- **region**

La región en la que reside la macro.

- **accountId**

El ID de la cuenta desde la que la macro invoca la función de Lambda.

- **fragment**

El contenido de la plantilla disponible para el procesamiento personalizado, en formato JSON.

- Para las macros incluidas en la sección de plantilla `Transform`, esto es toda la plantilla, a excepción de la sección `Transform`.
- Para macros incluidas en una llamada de función intrínseca `Fn::Transform`, esto incluye todos los nodos secundarios (y sus hijos) en función de la ubicación de la función intrínseca dentro de la plantilla, excepto para la función `Fn::Transform`. Para obtener más información, consulte [Alcance de la plantilla de macro](#).

- **transformId**

El nombre de la macro que invoca esta función.

- **params**

Para las llamadas de la función `Fn::Transform`, todos los parámetros especificados para la función. CloudFormation no evalúa estos parámetros antes de transferirlos a la función.

Para macros incluidas en la sección de plantilla `Transform`, esta sección está vacía.

- `requestId`

El ID de la solicitud que invoca esta función.

- `templateParameterValues`

Todos los parámetros especificados en la sección [Parameters](#) de la plantilla. CloudFormation evalúa estos parámetros antes de transferirlos a la función.

Formato de respuesta

CloudFormation espera que la función de Lambda devuelva una respuesta con el siguiente formato JSON:

```
{
  "requestId" : "$REQUEST_ID",
  "status" : "$STATUS",
  "fragment" : { ... },
  "errorMessage": "optional error message for failures"
}
```

- `requestId`

El ID de la solicitud que invoca esta función. Debe coincidir con el ID de la solicitud que proporciona CloudFormation al invocar la función.

- `status`

El estado de la solicitud (no distingue entre mayúsculas y minúsculas). Debe definirse en `success`. CloudFormation trata todas las demás respuestas como error.

- `fragment`

El contenido de la plantilla procesada para que CloudFormation lo incluya en la plantilla procesada, incluidos los elementos secundarios. CloudFormation sustituye el contenido de la plantilla que se transfiere a la función de Lambda con el fragmento de código de plantilla que recibe en la respuesta de Lambda.

El contenido de la plantilla procesada tiene que ser un formato JSON válido, y su inclusión en la plantilla procesada debe generar una plantilla válida.

Si la función no cambia realmente el contenido de la plantilla que CloudFormation le pasa, pero el usuario sigue necesitando incluir ese contenido en la plantilla procesada, la función debe devolver dicho contenido de plantilla a CloudFormation en su respuesta.

- `errorMessage`

El mensaje de error que explica por qué falló la transformación. CloudFormation muestra este mensaje de error en el panel Events (Eventos) de la página Stack details (Detalles de la pila) para la pila.

Por ejemplo:

```
Error creating change set: Transform
      Cuenta de AWS account
      number::macro name failed with:
      error message string.
```

Creación de una definición de macro

Crear una definición de macro de CloudFormation

1. [Cree una función de Lambda](#) que se encargue del procesamiento del contenido de la plantilla. Puede procesar cualquier parte de una plantilla, incluso la plantilla completa.
2. Cree una plantilla de CloudFormation que contenga un tipo de recurso `AWS::CloudFormation::Macro` y especifique las propiedades `Name` y `FunctionName`. La propiedad `FunctionName` debe contener el ARN de la función de Lambda para invocar cuándo CloudFormation ejecuta la macro.
3. (Opcional) Para ayudar en la depuración, también puede especificar las propiedades `LogGroupName` y `LogRoleArn` cuando cree el tipo de recurso `AWS::CloudFormation::Macro` para su macro. Estas propiedades le permiten especificar el grupo de registro de Registros de CloudWatch al que CloudFormation envía la información de registro de errores al invocar la función de Lambda subyacente de la macro y el rol que CloudFormation debe asumir al enviar entradas de registro a dichos registros.
4. [Cree una pila](#) utilizando la plantilla con la macro de la cuenta en la que desee utilizarla. O [cree un conjunto de pilas con permisos autoadministrados](#) mediante la plantilla con la macro en la cuenta de administrador y, a continuación, cree instancias de pilas en las cuentas de destino.

5. Una vez que CloudFormation haya creado correctamente las pilas que contienen la definición de la macro, esta estará disponible para su uso en dichas cuentas. Una macro se utiliza haciendo referencia a ella en una plantilla, en la ubicación adecuada relevante para el contenido de la plantilla que desea procesar.

Alcance de la plantilla de macro

Las macros a las que se hace referencia en la sección `Transform` de una plantilla pueden procesar todo el contenido de dicha plantilla.

Las macros a las que se hace referencia en una función `Fn::Transform` pueden procesar el contenido de cualquiera de los elementos del mismo nivel (incluidos los elementos secundarios) de dicha función `Fn::Transform` en la plantilla.

Por ejemplo, en la siguiente plantilla de ejemplo, `AWS::Include` puede procesar todas las propiedades `MyBucket`, en función de la ubicación de la función `Fn::Transform` que lo contiene. `MyMacro` puede procesar el contenido de toda la plantilla debido a su inclusión en la sección `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
```

Orden de evaluación de macro

Puede consultar varias macros de una determinada plantilla, incluidas transformaciones alojadas por CloudFormation, como [AWS::Include](#) y [AWS::Serverless](#).

Las macros se evalúan en orden, en función de su ubicación en la plantilla, desde la que esté más profundamente anidada hasta la más general. Las macros que están en la misma ubicación de la plantilla se evalúan en serie en función del orden en que aparecen en la lista.

Las transformaciones como, por ejemplo, `AWS::Include` y `AWS::Transform`, se tratan de la misma manera que cualquier otra macro en términos de orden y ámbito de acción.

Por ejemplo, en la siguiente plantilla de ejemplo, CloudFormation evalúa primero la macro `PolicyAdder`, ya que es la macro que está anidada más profundamente en la plantilla. A continuación, CloudFormation evalúa `MyMacro` antes de evaluar `AWS::Serverless`, ya que aparece antes de `AWS::Serverless` en la sección `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
```

Ejemplo de macro sencilla de reemplazo de cadenas

En el siguiente ejemplo, se explica con detalle el proceso de utilización de las macros, desde la definición de la macro en una plantilla hasta la creación de una función de Lambda para la macro y, a continuación, la utilización de la macro en una plantilla.

En este ejemplo, creamos un macro sencilla que inserta la cadena especificada en lugar del contenido de destino especificado en la plantilla procesada. Después la utilizaremos para insertar una `WaitHandleCondition` en blanco en la ubicación especificada en la plantilla procesada.

Creación de una macro

Antes de utilizar una macro, primero hay que hacer dos cosas: crear la función de Lambda que se encarga del procesamiento de la plantilla y poner la función de Lambda a disposición de CloudFormation creando una definición de macro.

La plantilla de ejemplo siguiente contiene la definición de nuestra macro de ejemplo. Para que la macro esté disponible en una Cuenta de AWS específica, tenemos que crear una pila a partir de la plantilla. La definición de la macro especifica el nombre de la macro, una breve descripción y hace referencia al ARN de la función de Lambda que CloudFormation invoca cuando esta macro se utiliza en una plantilla. (No hemos incluido una propiedad `LogGroupName` o `LogRoleARN` para el registro de errores).

En este ejemplo se presupone que la pila creada a partir de esta plantilla se denomina `JavaMacroFunc`. Debido a que la propiedad `Name` de la macro se establece en el nombre de la pila, la macro resultante también se denomina `JavaMacroFunc`.

```
AWS::FormatVersion: 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'
```

Uso de la macro

Para utilizar nuestra macro, la incluiremos en una plantilla utilizando la función intrínseca `Fn::Transform`.

Cuando creamos una pila utilizando la plantilla siguiente, CloudFormation llamará a nuestra macro de ejemplo. La función de Lambda subyacente sustituirá una cadena especificada por otra cadena especificada. En este caso, el resultado es que se inserta un `AWS::CloudFormation::WaitConditionHandle` en blanco en la plantilla procesada.

```
Parameters:
```

```
ExampleParameter:
  Type: String
  Default: 'SampleMacro'

Resources:
  2a:
    Fn::Transform:
      Name: "JavaMacroFunc"
      Parameters:
        replacement: 'AWS::CloudFormation::WaitConditionHandle'
        target: '$$REPLACEMENT$$'
      Type: '$$REPLACEMENT$$'
```

- La macro para invocar se especifica como `JavaMacroFunc`, que proviene del ejemplo de definición de macro anterior.
- Se pasan dos parámetros a la macro, `target` y `replacement`, que representan la cadena de destino y su valor de sustitución deseado.
- La macro puede funcionar con el contenido del nodo `Type` porque `Type` tiene el mismo nivel que la función `Fn::Transform` que hace referencia a la macro.
- El elemento `AWS::CloudFormation::WaitConditionHandle` obtenido se denomina `2a`.
- La plantilla también contiene un parámetro de plantilla, `ExampleParameter`, al que la macro también tiene acceso (pero no utiliza en este caso).

Datos de entrada de Lambda

Cuando CloudFormation procesa nuestra plantilla de ejemplo durante la creación de la pila, pasa el siguiente mapeo de eventos a la función de Lambda a la que se hace referencia en la definición de la macro `JavaMacroFunc`.

- `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 contiene JSON que representa el fragmento de plantilla que la macro puede procesar. Este fragmento consta de los elementos de la llamada de función `Fn::Transform`, pero no la llamada de función en sí misma. Además, `params` contiene JSON que representa los parámetros de la macro. En este caso, la sustitución y el destino. Del mismo modo, `templateParameterValues` contiene JSON que representa los parámetros que se han especificado para la plantilla en su totalidad.

Código de la función de Lambda

A continuación se muestra el código real de la función de Lambda que subyace a la macro de ejemplo `JavaMacroFunc`. Se itera en el fragmento de código de plantilla incluido en la respuesta (ya sea en formato de cadena, lista o mapa), buscando la cadena de destino especificada. Si encuentra la cadena de destino especificada, la función de Lambda sustituye la cadena de destino por la cadena de sustitución especificada. Si no es así, la función deja el fragmento de código de plantilla sin cambiar. A continuación, la función devuelve un mapa de las propiedades esperadas, que se describen en detalle a continuación, en 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;
}
}
```

Respuesta de la función de Lambda

A continuación, se presenta la asignación que la función de Lambda devuelve a CloudFormation para su procesamiento.

- `requestId` : 5dba79b5-f117-4de0-9ce4-d40363bfb6ab
- `status` : SUCCESS
- `fragment` :

```
{
  "Type": "AWS::CloudFormation::WaitConditionHandle"
}
```

El `requestId` coincide con el que se envió desde CloudFormation y un valor `status` de SUCCESS indica que la función de Lambda ha procesado correctamente el fragmento de código de plantilla incluido en la solicitud. En esta respuesta, `fragment` contiene JSON que representa el contenido que debe insertarse en la plantilla procesada en lugar del fragmento de código de plantilla original.

Plantilla procesada resultante

Una vez que CloudFormation recibe una respuesta correcta de la función de Lambda, se inserta el fragmento de código de plantilla devuelto en la plantilla procesada.

A continuación se muestra la plantilla procesada resultante para nuestro ejemplo. La llamada de función intrínseca `Fn::Transform` a la que hace referencia la macro `JavaMacroFunc` ya no está incluida. El fragmento de plantilla devuelto por la función de Lambda se incluye en la ubicación

adecuada, con el resultado de que el contenido "Type": "\$\$REPLACEMENT\$\$" se ha sustituido con "Type": "AWS::CloudFormation::WaitConditionHandle".

```
{
  "Parameters": {
    "ExampleParameter": {
      "Default": "SampleMacro",
      "Type": "String"
    }
  },
  "Resources": {
    "2a": {
      "Type": "AWS::CloudFormation::WaitConditionHandle"
    }
  }
}
```

Solución de problemas de la plantilla procesada

Cuando se utiliza una macro, la plantilla procesada se encuentra en la consola de CloudFormation.

La etapa de una plantilla indica su estado de procesamiento:

- **Original:** la plantilla que el usuario envió originalmente para crear o actualizar la pila o el conjunto de pilas.
- **Processed:** la plantilla que CloudFormation utilizó para crear o actualizar la pila o el conjunto de pilas después de procesar todas las macros de referencia. La plantilla procesada se formatea como JSON, incluso si la plantilla original tenía el formato YAML.

Utilice la plantilla procesada para la resolución de problemas. Si una plantilla no hace referencia a macros, las plantillas original y procesada son idénticas.

Para obtener más información, consulte [Visualización de la información de la pila desde la consola de CloudFormation](#).

Para utilizar la AWS CLI para obtener la plantilla procesada, utilice el comando [get-template](#).

Límites de tamaño

El tamaño máximo de una plantilla de pila procesada es de 51 200 bytes cuando se pasa directamente a una solicitud CreateStack, UpdateStack o ValidateTemplate, o 1 MB cuando

se pasa como un objeto S3 utilizando una URL de plantilla de Amazon S3. Sin embargo, durante el procesamiento, CloudFormation actualiza el estado temporal de la plantilla, ya que procesa en serie las macros contenidas en la plantilla. Por este motivo, el tamaño de la plantilla durante el procesamiento puede superar temporalmente el tamaño máximo permitido de una plantilla totalmente procesada. CloudFormation permite cierta cantidad de búfer para estas plantillas en proceso. Sin embargo, debería diseñar las plantillas y macros teniendo en cuenta el tamaño máximo permitido para una plantilla de pila procesada.

Si CloudFormation devuelve un error `Transformation data limit exceeded` al procesar su plantilla, significa que la plantilla ha superado el tamaño máximo de plantilla que CloudFormation permite durante el procesamiento.

Para resolver este problema, realice lo siguiente:

- Reestructure la plantilla en varias plantillas para evitar superar el tamaño máximo de las plantillas en proceso. Por ejemplo:
 - Utilice plantillas de pila anidadas para encapsular las partes de la plantilla. Para obtener más información, consulte [División de una plantilla en piezas reutilizables utilizando pilas anidadas](#).
 - Cree varias pilas y utilice referencias de pila cruzadas para intercambiar información entre ellas. Para obtener más información, consulte [Consulta de las salidas de recursos en otra pila de CloudFormation](#).
- Reduzca el tamaño de fragmento de plantilla devuelto por un determinado macro. CloudFormation no modifica el contenido de los fragmentos devuelto por macros.

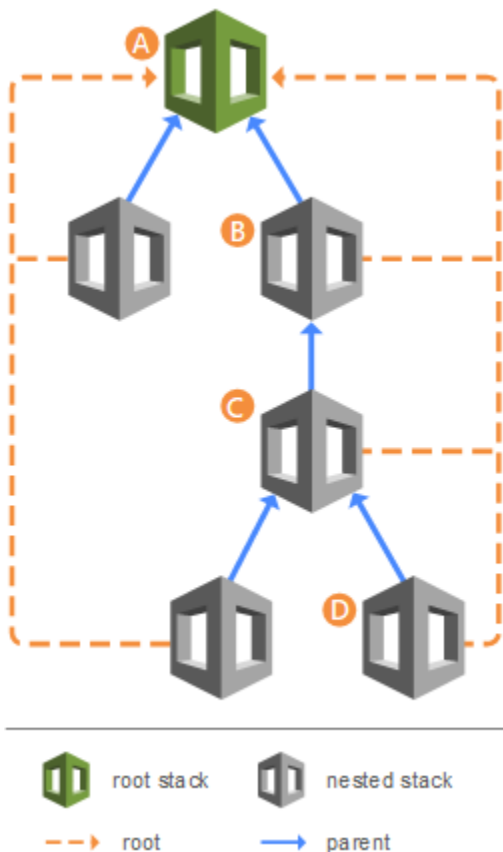
División de una plantilla en piezas reutilizables utilizando pilas anidadas

A medida que su infraestructura crezca, es posible que se encuentre creando repetidamente configuraciones de recursos idénticas en varias plantillas. Para evitar esta redundancia, puede separar estas configuraciones comunes en plantillas dedicadas. Después, puede utilizar el recurso [AWS::CloudFormation::Stack](#) en otras plantillas para hacer referencia a otras plantillas dedicadas, y así crear pilas anidadas.

Por ejemplo, imagine que tiene una configuración de equilibrador de carga que utiliza para la mayoría de sus pilas. En lugar de copiar y pegar las mismas configuraciones en las plantillas, puede crear una plantilla exclusiva para el balanceador de carga. Luego, puede hacer referencia a esta plantilla desde otras plantillas que requieran la misma configuración de equilibrador de carga.

Las pilas anidadas pueden a su vez contener otras pilas anidadas, lo que se traduce en una jerarquía de pilas, como se muestra en el diagrama que aparece a continuación. La pila raíz es la pila de nivel superior a la que pertenecen todas las demás pilas anidadas. Cada pila anidada dispone de una pila primaria inmediata. En el caso del primer nivel de pilas anidadas, la pila raíz es a su vez la primaria.

- La pila A es la raíz de todas las demás pilas anidadas de la jerarquía.
- En el caso de la pila B, la pila A es tanto su pila primaria como la pila raíz.
- Para la pila D, la C es su pila primaria, del mismo modo que para la pila C, la B es su pila primaria.



Temas

- [Ejemplo de antes y después de dividir una plantilla](#)
- [Ejemplo de una arquitectura de pila anidada](#)
- [Operaciones de apilamiento en pilas anidadas](#)
- [Información relacionada](#)

Ejemplo de antes y después de dividir una plantilla

En este ejemplo, se muestra cómo puede tomar una plantilla grande y única de CloudFormation y reorganizarla en un diseño más estructurado y reutilizable mediante plantillas anidadas. Inicialmente, la plantilla “Antes de anidar las pilas” muestra todos los recursos definidos en un archivo. Esto puede resultar complicado y difícil de administrar a medida que aumenta la cantidad de recursos. La plantilla «Después de anidar las pilas» divide los recursos en plantillas más pequeñas e independientes. Cada pila anidada gestiona un conjunto específico de recursos relacionados, lo que hace que la estructura general esté más organizada y sea más fácil de mantener.

Antes de anidar las pilas	Después de anidar las pilas
<pre>AWSTemplateFormatVersion: 2010-09-09 Parameters: InstanceType: Type: String Default: t2.micro Description: The EC2 instance type Environment: Type: String Default: Production Description: The deployment environment Resources: MyEC2Instance: Type: AWS::EC2::Instance Properties: ImageId: ami-1234567890abcdef0 InstanceType: !Ref InstanceType MyS3Bucket: Type: AWS::S3::Bucket</pre>	<pre>AWSTemplateFormatVersion: 2010-09-09 Resources: MyFirstNestedStack: Type: AWS::CloudFormation::Stack Properties: TemplateURL: https://s3.amazonaws.com/ amzn-s3-demo-bucket /first-nested-stack.yaml Parameters: # Pass parameters to the nested stack if needed InstanceType: t3.micro MySecondNestedStack: Type: AWS::CloudFormation::Stack Properties: TemplateURL: https://s3.amazonaws.com/ amzn-s3-demo-bucket /second-nested-stack.yaml Parameters: # Pass parameters to the nested stack if needed Environment: Testing DependsOn: MyFirstNestedStack</pre>

Ejemplo de una arquitectura de pila anidada

Esta sección muestra una arquitectura de pila anidada que consiste en una pila de nivel superior que hace referencia a una pila anidada. La pila anidada implementa una función de Lambda de Node.js, recibe un valor de parámetro de la pila de nivel superior y devuelve una salida que se expone a través de la pila de nivel superior.

Temas

- [Paso 1: cree una plantilla para la pila anidada en su sistema local.](#)
- [Paso 2: cree una plantilla para la pila de nivel superior en su sistema local.](#)
- [Paso 3: empaquetar e implementar las plantillas.](#)

Paso 1: cree una plantilla para la pila anidada en su sistema local.

En el siguiente ejemplo, se muestra el formato de la plantilla para la pila anidada.

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

```

Paso 2: cree una plantilla para la pila de nivel superior en su sistema local.

El siguiente ejemplo muestra el formato de la plantilla de pila de nivel superior y el recurso [AWS::CloudFormation::Stack](#) que hace referencia a la pila que creó en el paso anterior.

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

```

Paso 3: empaquetar e implementar las plantillas.

Note

Al trabajar con plantillas de forma local, el comando `package` de la AWS CLI puede ayudarlo a preparar las plantillas para su implementación. Gestiona automáticamente la carga de artefactos locales en Amazon S3 (incluidas `TemplateURL`) y genera un nuevo archivo de plantilla con referencias actualizadas a estas ubicaciones de S3. Para obtener más información, consulte [Carga de artefactos locales en un bucket de S3 con la AWS CLI](#).

A continuación, puede utilizar el comando `package` para cargar la plantilla anidada a un bucket de 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
```

El comando genera una nueva plantilla en la ruta especificada por `--output-template-file`. Reemplaza la referencia `TemplateURL` con la ubicación de Amazon S3, tal y como se muestra a continuación.

Plantilla resultante

```
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
```

Después de que ejecute el comando `package`, puede implementar la plantilla procesada mediante el comando [deploy](#). Para las pilas anidadas que contienen recursos de IAM, debe confirmar las capacidades de IAM al incluir la opción `--capabilities`.

```
aws cloudformation deploy \  
  --template-file packaged-template.yaml \  
  --stack-name stack-name \  
  --capabilities CAPABILITY_NAMED_IAM
```

Operaciones de apilamiento en pilas anidadas

Al trabajar con pilas anidadas, debe gestionarlas con cuidado durante las operaciones. Determinadas operaciones de las pilas, como las actualizaciones de pilas, deben iniciarse en la pila raíz en lugar de realizarse directamente en las pilas anidadas. Al actualizar una pila raíz, solo se actualizarán las pilas anidadas con cambios de plantilla.

Además, la presencia de las pilas anidadas puede afectar a las operaciones en la pila raíz. Por ejemplo, si una pila anidada se queda atascada en el estado `UPDATE_ROLLBACK_IN_PROGRESS`, la pila raíz esperará hasta que esa pila anidada complete su proceso de reversión antes de continuar. Antes de continuar con las operaciones de actualización, asegúrese de tener permisos de IAM para cancelar una actualización de la pila en caso de que se revierta. Para obtener más información, consulte [Control del acceso a CloudFormation con AWS Identity and Access Management](#).

Use los siguientes procedimientos para buscar la pila raíz y las pilas anidadas.

Para ver la pila raíz de una pila anidada

1. Inicie sesión en la Consola de administración de AWS y abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation>.
2. En la página Pilas, seleccione el nombre de la pila anidada de la que desea ver la pila raíz.

Las pilas anidadas tienen la palabra ANIDADA sobre su nombre.

3. En la pestaña Información de la pila, en la sección Información general, seleccione el nombre de la pila mostrada como Pila raíz.

Para ver las pilas anidadas que pertenecen a una pila raíz

1. Desde la pila raíz cuyas pilas anidadas desea ver, seleccione la pestaña Recursos.
2. En la columna Tipo, busque recursos del tipo `AWS::CloudFormation::Stack`.

Información relacionada

- [Anidamiento de una pila existente](#)
- [Comprensión de los comportamientos de actualización de los recursos de la pila](#)
- [Continúe con la reversión de las actualizaciones con fallas de la pila anidada](#)
- [Fallo de reversión de pilas anidadas](#)

Creación de condiciones de espera en una plantilla de CloudFormation

En este tema se explica cómo crear una condición de espera en una plantilla para coordinar la creación de los recursos de la pila o realizar un seguimiento del progreso de un proceso de configuración. Por ejemplo, puede comenzar la creación de otro recurso después de que la configuración de la aplicación esté parcialmente completa o puede enviar señales durante un proceso de instalación y configuración para realizar un seguimiento de su progreso.

Cuando CloudFormation crea una pila que incluye una condición de espera:

- Crea una condición de espera igual que cualquier otro recurso y establece el estado de la condición de espera en `CREATE_IN_PROGRESS`.
- CloudFormation espera hasta que recibe la cantidad necesaria de señales de éxito o hasta que haya vencido el periodo de la condición de espera.
- Si recibe la cantidad requerida de señales de éxito antes de que venza el periodo de espera:
 - El estado de la condición de espera cambia a `CREATE_COMPLETE`
 - La creación de la pila continúa
- Si se agota el tiempo de espera o se recibe una señal de error:
 - El estado de la condición de espera cambia a `CREATE_FAILED`
 - Restauración de una pila

 Important

Con los recursos de Amazon EC2 y Auto Scaling recomendamos que use un atributo `CreationPolicy` en lugar de condiciones de espera. Agregue un atributo `CreationPolicy` a esos recursos y use el script de ayuda `cf-signal` para señalar que el proceso de creación de instancias ha finalizado satisfactoriamente.

Para obtener más información, consulte [CreationPolicy atributo](#).

 Note

Si utiliza AWS PrivateLink, los recursos de la VPC que responden a condiciones de espera tienen que tener acceso a buckets de Amazon Simple Storage Service (Amazon S3) específicos de CloudFormation. Los recursos deben enviar respuestas de condición de espera a una URL de Amazon S3 prefirmada. Si no pueden enviar respuestas a Amazon S3, CloudFormation no recibirá una respuesta y la operación de pila producirá un error.

Para obtener más información, consulte [Acceso a CloudFormation a través de un punto de conexión de la interfaz \(AWS PrivateLink\)](#) y [Control del acceso desde puntos de conexión de VPC con políticas de bucket](#).

Temas

- [Creación de una condición de espera en su plantilla](#)
- [Sintaxis de señal de condición de espera](#)
- [Acceso a los datos de señales](#)

Creación de una condición de espera en su plantilla

1. Identificador de condición de espera

Se empieza por definir un recurso `AWS::CloudFormation::WaitConditionHandle` en la plantilla de la pila. Este recurso genera la URL prefirmada necesaria para enviar señales. Esto le permite enviar una señal sin tener que proporcionar sus credenciales de AWS. Por ejemplo:

Resources:

MyWaitHandle:

Type: `AWS::CloudFormation::WaitConditionHandle`

2. Condición de espera

A continuación, debe definir un recurso [AWS::CloudFormation::WaitCondition](#) en la plantilla de la pila. La estructura básica de `AWS::CloudFormation::WaitCondition` tiene un aspecto similar al siguiente:

```
MyWaitCondition:
  Type: AWS::CloudFormation::WaitCondition
  Properties:
    Handle: String
    Timeout: String
    Count: Integer
```

El recurso `AWS::CloudFormation::WaitCondition` tiene dos propiedades obligatorias y una propiedad opcional.

- **Handle** (obligatoria): referencia a un `WaitConditionHandle` declarado en la plantilla.
- **Timeout** (obligatoria): la cantidad de segundos que CloudFormation debe esperar a que se reciba la cantidad de señales requerida. `Timeout` es una propiedad con un límite mínimo, lo que quiere decir que el tiempo de espera comienza a partir del momento en que se especifique, no antes, aunque también puede producirse poco después. El tiempo máximo que puede especificar es de 43200 segundos (12 horas).
- **Count** (opcional): la cantidad de señales de éxito que CloudFormation debe recibir antes de configurar el estado de esa condición de espera en `CREATE_COMPLETE` y retoma la creación de la pila. Si no se especifica, el valor predeterminado es 1.

Normalmente, querrá que una condición de espera comience inmediatamente después de la creación de un recurso específico. Para ello, agregue el atributo `DependsOn` a una condición de espera. Al añadir un atributo `DependsOn` a una condición de espera, CloudFormation crea primero el recurso en el atributo `DependsOn` y, a continuación, crea la condición de espera. Para obtener más información, consulte [DependsOn atributo](#).

En el siguiente ejemplo se muestra una condición de espera que:

- Comienza después de la creación correcta del recurso `MyEC2Instance`
- Utiliza el recurso `MyWaitHandle` como `WaitConditionHandle`

- Tiene un tiempo de espera de 4500 segundos
- Tiene el valor predeterminado Count de 1 (ya que no se especifica ninguna propiedad Count)

MyWaitCondition:

```
Type: AWS::CloudFormation::WaitCondition
DependsOn: MyEC2Instance
Properties:
  Handle: !Ref MyWaitHandle
  Timeout: '4500'
```

3. Envío de una señal

Para indicar el éxito o el error de CloudFormation, normalmente se ejecuta algún código o script. Por ejemplo, una aplicación que se ejecuta en una instancia de EC2 puede realizar algunas tareas de configuración adicionales y, a continuación, enviar una señal a CloudFormation para indicar que ha finalizado.

La señal debe enviarse a la URL prefirmada generada por el identificador de la condición de espera. Debe utilizar la URL prefirmada para señalar el éxito o el error.

Enviar una señal

1. Para recuperar la URL prefirmada dentro de la plantilla, utilice la función intrínseca Ref con el nombre lógico del identificador de la condición de espera.

Como se muestra en el siguiente ejemplo, su plantilla puede indicar una instancia de Amazon EC2 y pasar la URL prefirmada a las instancias de EC2 mediante la propiedad UserData de Amazon EC2. Esto permite que los scripts o las aplicaciones que se ejecutan en esas instancias indiquen el éxito o el error a 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" }
```

Esto devuelve un resultado UserData similar a:

```
SignalURL=https://amzn-s3-demo-bucket.s3.amazonaws.com/...
```

Nota: En la Consola de administración de AWS o las herramientas de línea de comandos, la URL prefirmada se muestra con el ID físico de la condición de espera del recurso del administrador.

2. (Opcional) Para detectar cuándo la pila entra en estado de espera, puede usar uno de los siguientes métodos:
 - Si crea la pila con las notificaciones habilitadas, CloudFormation publica una notificación para cada evento de pila en el tema especificado. Si usted o su aplicación se suscriben a ese tema, puede monitorizar las notificaciones para el evento de creación del identificador de la condición de espera y recuperar la URL prefirmada del mensaje de notificación.
 - También puede monitorear los eventos de la pila mediante la Consola de administración de AWS, la AWS CLI o un SDK.
3. Para enviar una señal, envía un mensaje de solicitud HTTP usando la URL prefirmada. El método de solicitudes debe ser PUT y el encabezado Content-Type debe ser una cadena vacía u omitirse. El mensaje de solicitud debe ser una estructura JSON con la forma especificada en [Sintaxis de señal de condición de espera](#).

Debe enviar la cantidad de señales de éxito que se especifica en la propiedad Count para que CloudFormation continúe con la creación de la pila. Si la propiedad Count es superior a 1, el valor UniqueId para cada señal debe ser único entre todas las señales enviadas a una condición de espera en particular. El UniqueId es una cadena alfanumérica arbitraria.

Un comando `curl` es una forma de enviar una señal. El siguiente ejemplo muestra una línea de comandos `curl` que señala el éxito a una condición de espera.

```
$ 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"
```

donde el archivo */tmp/a* contiene la siguiente estructura JSON:

```
{
  "Status" : "SUCCESS",
  "Reason" : "Configuration Complete",
  "UniqueId" : "ID1234",
  "Data" : "Application has completed configuration."
}
```

En este ejemplo se muestra una línea de comandos `curl` que envía la misma señal de éxito, salvo que envía la estructura JSON como parámetro en la línea de comandos.

```
$ 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"
```

Sintaxis de señal de condición de espera

Cuando envía señales a la URL generada por el administrador de la condición de espera, tiene que utilizar el siguiente formato JSON:

```
{
  "Status" : "StatusValue",
  "UniqueId" : "Some UniqueId",
  "Data" : "Some Data",
  "Reason" : "Some Reason"
}
```

Propiedades

El campo `Status` debe tener uno de los siguientes valores:

- SUCCESS

- FAILURE

El campo `UniqueId` identifica la señal a CloudFormation. Si la propiedad `Count` de la condición de espera es superior a 1, el valor `UniqueId` debe ser único en todas las señales enviadas para una determinada condición de espera; de lo contrario, CloudFormation considerará que la señal es una retransmisión de la señal enviada con anterioridad con el mismo `UniqueId` y la ignorará.

El campo `Data` puede contener cualquier información que desea devolver con la señal. Puede acceder al valor `Data` usando la función [Fn::GetAtt](#) dentro de la plantilla.

El campo `Reason` es una cadena sin ninguna restricción en su contenido, salvo la conformidad de JSON.

Acceso a los datos de señales

Para acceder a los datos enviados por señales válidas, puede crear un valor de salida para la condición de espera en la plantilla de CloudFormation. Por ejemplo:

Outputs:

WaitConditionData:

Description: *The data passed back as part of signalling the WaitCondition*

Value: `!GetAtt MyWaitCondition.Data`

Luego, puede ver estos datos con el comando [describe-stacks](#) o la pestaña Salida de la consola de CloudFormation.

La función `Fn::GetAtt` devuelve los valores `UniqueId` y `Data` como un par nombre-valor dentro de una estructura JSON. Por ejemplo:

```
{"Signal1": "Application has completed configuration."}
```

Creación de configuraciones de recursos reutilizables que se puedan incluir en las plantillas con los módulos de CloudFormation

Los módulos son una forma de empaquetar configuraciones de recursos para su inclusión en plantillas de pila, de forma transparente, administrable y repetible. Los módulos pueden encapsular configuraciones de servicios comunes y prácticas recomendadas como componentes esenciales modulares y personalizables para que pueda incluirlos en las plantillas de pila. Los módulos le

permiten incluir configuraciones de recursos que incorporan prácticas recomendadas, conocimientos de dominio experto y directrices aceptadas en sus plantillas (para áreas como seguridad, cumplimiento, gestión y regulaciones del sector), sin tener que adquirir un conocimiento profundo de las complejidades de la implementación de recursos.

Por ejemplo, un experto en redes podría crear un módulo que contenga grupos de seguridad integrados y reglas de entrada/salida que cumplan con las directrices de seguridad. A continuación, podría incluir ese módulo en la plantilla para aprovisionar una infraestructura de red segura en su pila, sin tener que dedicar tiempo a averiguar cómo funcionan las VPC, las subredes, los grupos de seguridad y las gateway. Y dado que los módulos están versionados, si las pautas de seguridad cambian con el tiempo, el autor del módulo podría crear una nueva versión del módulo que incorpore esos cambios.

Las características del uso de módulos en sus plantillas incluyen:

- **Previsibilidad:** un módulo debe cumplir con el esquema que registra en el registro de CloudFormation, para que sepa en qué recursos puede resolver una vez que lo incluya en la plantilla.
- **Reutilización:** puede usar el mismo módulo en varias plantillas y cuentas.
- **Trazabilidad:** CloudFormation retiene el conocimiento de qué recursos de una pila se aprovisionaron desde un módulo, lo que le permite comprender fácilmente el origen de los cambios en los recursos.
- **Capacidad de administración:** una vez registrado un módulo, puede administrarlo a través del registro de CloudFormation, incluido el control de versiones y la disponibilidad de cuentas y regiones.

Un módulo puede contener:

- Uno o más recursos que se aprovisionarán desde el módulo, junto con cualquier dato asociado, como salidas o condiciones.
- Cualquier parámetro del módulo, que le permite especificar valores personalizados cada vez que se utiliza el módulo.

Para obtener información sobre el desarrollo de módulos, consulte [Desarrollo de módulos](#) en la Guía del usuario de CLI de CloudFormation.

Temas

- [Consideraciones sobre el uso de módulos](#)
- [Comprensión del control de versiones del módulo](#)
- [Uso de módulos del registro privado de CloudFormation](#)
- [Uso de parámetros para especificar valores de módulo](#)
- [Recursos del módulo de referencia en las plantillas de CloudFormation](#)

Consideraciones sobre el uso de módulos

- El uso de los módulos no conlleva ningún coste adicional. Usted paga sólo por los recursos a los que esos módulos resuelven en sus pilas.
- Las cuotas de CloudFormation, como el número máximo de recursos permitidos en una pila o el tamaño máximo del cuerpo de la plantilla, se aplican a la plantilla procesada independientemente de si los recursos incluidos en esa plantilla provienen de módulos o no. Para obtener más información, consulte [Comprensión de las cuotas de CloudFormation](#).
- Las etiquetas que especifique en el nivel de pila se asignan a los recursos individuales derivados del módulo.
- Los scripts auxiliares especificados en el nivel de módulo no se propagan a los recursos individuales contenidos en el módulo cuando CloudFormation procesa la plantilla.
- Las salidas especificadas en el módulo se propagan a las salidas a nivel de plantilla.

A cada salida se le asignará un ID lógico, que es una concatenación del nombre lógico del módulo y el nombre de salida tal como se define en el módulo. Para obtener más información, consulte [Obtención de salidas exportadas de una pila implementada de CloudFormation](#).

- Los parámetros especificados en el módulo no se propagan a parámetros a nivel de plantilla.

Sin embargo, puede crear parámetros de nivel de plantilla que hagan referencia a parámetros de nivel de módulo. Para obtener más información, consulte [Uso de parámetros para especificar valores de módulo](#).

Comprensión del control de versiones del módulo

El registro de CloudFormation actúa como un repositorio en el que puede registrar y administrar los módulos para usarlos en su Cuenta de AWS y en su región. Puede registrar módulos de diversas fuentes, como publicadores de terceros de AWS y sus propias extensiones personalizadas en su

cuenta y región. Para obtener más información, consulte [Administración de extensiones a través del registro de CloudFormation](#).

Los módulos pueden tener diferentes versiones, por lo que puede especificar qué versión del módulo desea utilizar. Esta capacidad de control de versiones es especialmente útil cuando se necesita actualizar o modificar un módulo sin interrumpir las pilas existentes que dependen de él.

Tenga en cuenta las siguientes consideraciones al utilizar varias versiones de un módulo:

- Durante las operaciones de pila, CloudFormation usa cualquier versión del módulo que esté actualmente registrada como la versión predeterminada en la Cuenta de AWS y región en la que se realice la operación de pila. Esto incluye módulos anidados en otros módulos.

Por lo tanto, tenga en cuenta que si tiene diferentes versiones del mismo módulo registradas como la versión predeterminada en diferentes cuentas o regiones, el uso de la misma plantilla puede dar lugar a resultados diferentes.

- Durante las operaciones de pila, CloudFormation usa cualquier versión del recurso que esté actualmente registrada como la versión predeterminada en la Cuenta de AWS y región en la que se realice la operación de pila. Esto incluye los recursos generados mediante la inclusión de módulos.
- El cambio de la versión predeterminada de un módulo no inicia ninguna operación de actualización de pila. Sin embargo, la próxima vez que realice una operación de pila con cualquier plantilla que contenga ese módulo, como una actualización de pila, CloudFormation utilizará la nueva versión predeterminada en la operación.

La única excepción a esto es realizar una actualización de pila con la opción usar plantilla anterior especificada, como se describe a continuación.

- Para las operaciones de actualización de pila, si especifica la opción usar plantilla anterior, CloudFormation utiliza la plantilla procesada anterior para la actualización de la pila y no vuelve a procesar el módulo para los cambios que pueda haber realizado en él.
- Para garantizar resultados uniformes, si incluye módulos en una plantilla de pila para su uso con conjuntos de pilas, debe asegurarse de que la misma versión del módulo esté establecida como la versión predeterminada en todas las cuentas y regiones en las que planea implementar las instancias de pila. Esto incluye los módulos que están anidados en otros módulos. Para obtener más información, consulte [Administración de pilas entre cuentas y regiones con StackSets](#).

Requisitos para activar módulos públicos de terceros

Para activar correctamente un módulo público de terceros en su cuenta y región, debe cumplirse lo siguiente para cada extensión pública de terceros (recurso o módulo) incluida en el módulo:

- Activación de la extensión: la extensión debe estar activada en la cuenta y la región en las que desee utilizarla. Para obtener más información, consulte [Utilización de extensiones públicas de terceros del registro de CloudFormation](#).
- Registro del alias: si la extensión del módulo usa un alias de nombre de tipo, la extensión debe estar registrada en su cuenta y región con el mismo alias de nombre de tipo. Para obtener más información, consulte [Utilización de alias para hacer referencia a extensiones](#).
- Compatibilidad de la versión: la versión de la extensión actualmente activada debe ser una de las versiones principales admitidas de esa extensión especificadas en el módulo.

Si no tiene activadas las extensiones públicas de terceros y las versiones de extensión correctas, se producirá un error en la operación de CloudFormation que enumera las extensiones o versiones que deben activarse antes de que el módulo se pueda activar correctamente.

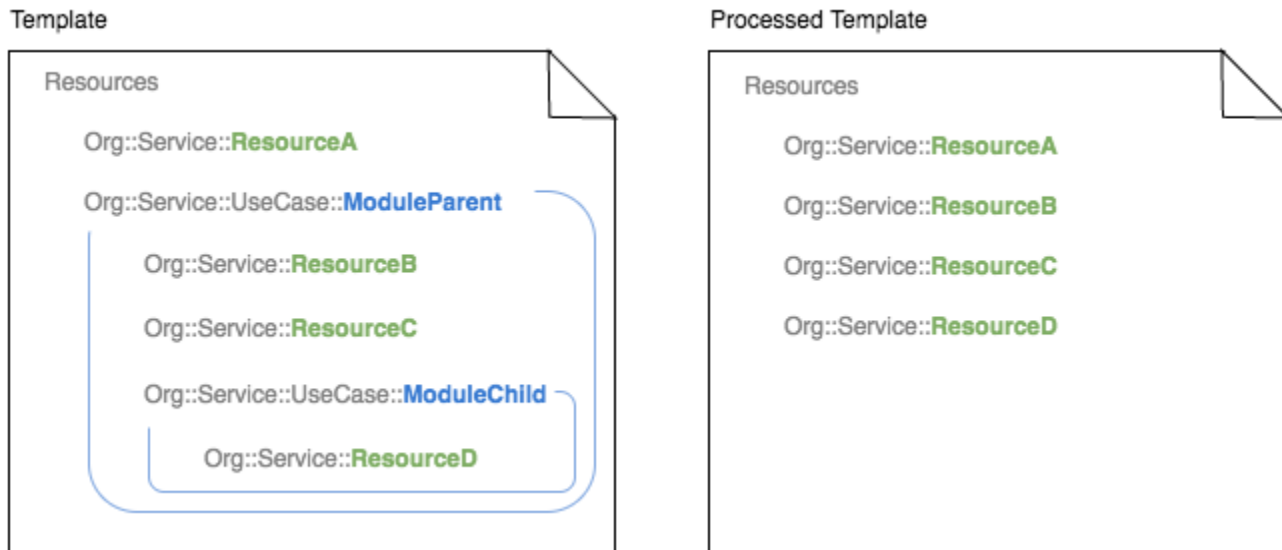
Uso de módulos del registro privado de CloudFormation

En este tema se explica cómo utilizar los módulos en las plantillas de CloudFormation. Piense en los módulos como agrupaciones de recursos prediseñados que puede agregar a sus plantillas.

Para usar un módulo, los pasos son los siguientes:

- Registre los módulos: registre los módulos en el registro de CloudFormation como extensiones privadas. Asegúrese de que esté registrado en la Cuenta de AWS y en la región en la que trabaja. Para obtener más información, consulte [Conceptos de registro de CloudFormation](#).
- Inclúyalo en la plantilla: agrega el módulo a la sección [Resources](#) de la plantilla de CloudFormation, tal como lo haría con otros recursos. También tendrá que proporcionar las propiedades necesarias para el módulo.
- Cree o actualice la pila: al iniciar una operación de pila, CloudFormation genera una plantilla procesada que resuelve los módulos incluidos en los recursos apropiados.
- Previsualice de los cambios: antes de realizar cambios, puede usar un conjunto de cambios para ver qué recursos se agregarán o cambiarán. Para obtener más información, consulte [Actualización de pilas de CloudFormation con conjuntos de cambios](#).

Considere el siguiente ejemplo: tiene una plantilla que contiene recursos y módulos. La plantilla contiene un recurso individual, `ResourceA`, como así también un módulo, `ModuleParent`. Ese módulo contiene dos recursos, `ResourceB` y `ResourceC`, además de un módulo anidado, `ModuleChild`. `ModuleChild` contiene un único recurso, `ResourceD`. Si crea una pila a partir de esta plantilla, CloudFormation procesa la plantilla y resuelve los módulos a los recursos apropiados. La pila resultante tiene cuatro recursos: `ResourceA`, `ResourceB`, `ResourceC` y `ResourceD`.



CloudFormation realiza un seguimiento de los recursos de una pila que se crearon a partir de módulos. Puede ver esta información en las pestañas Eventos, Recursos y Desviaciones de una pila determinada, y también se incluye en vistas previas de conjuntos de cambios.

Los módulos se distinguen de los recursos de una plantilla porque se adhieren a la siguiente convención de nomenclatura de cuatro partes, a diferencia de la típica convención de tres partes utilizada por los recursos:

```
organization::service::use-case::MODULE
```

Uso de parámetros para especificar valores de módulo

En CloudFormation, puede usar plantillas de parámetros para personalizar las pilas al proporcionar valores de entrada durante la creación o actualización de la pila. Estos parámetros le permiten cambiar ciertos aspectos de la pila en función de sus necesidades. Para obtener más información sobre cómo definir parámetros de plantilla, consulte [Sintaxis de Parameters de la plantilla de CloudFormation](#).

Del mismo modo, los módulos también pueden tener parámetros. Estos parámetros del módulo le permiten introducir valores personalizados en el módulo desde la plantilla (u otro módulo) que lo está utilizando. A continuación, el módulo puede utilizar estos valores personalizados para establecer los valores de propiedades de los recursos que contiene.

También puede definir parámetros de plantilla que establecen las propiedades del módulo, de modo que puedes introducir valores que se pasan al módulo en el momento de la operación de pila.

Si un módulo contiene un módulo anidado que tiene sus propios parámetros del módulo, puede:

- Especificar los valores de los parámetros del módulo anidado directamente en el módulo principal.
- Definir los parámetros correspondientes del módulo en el módulo principal que permitan que los parámetros del módulo anidado sean establecidos por la plantilla (o módulo) en la que está contenido el módulo principal.

Uso de parámetros de plantilla para especificar valores de parámetros del módulo

El siguiente ejemplo muestra cómo definir parámetros de plantilla que pasan valores a un módulo.

Esta plantilla que contiene `My::S3::SampleBucket::MODULE` define un parámetro de plantilla `BucketName`, que permite al usuario especificar un nombre de bucket de S3 durante la operación de pila.

```
# 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
```

Especificación de propiedades en recursos de un módulo secundario desde el módulo principal

El ejemplo siguiente ilustra cómo especificar valores de parámetro en un módulo anidado dentro de otro módulo.

Este primer módulo, `My::S3::SampleBucketPrivate::MODULE`, será el módulo secundario. Define dos parámetros: `BucketName` y `AccessControl`. Los valores especificados para estos parámetros se utilizan para especificar las propiedades `BucketName` y `AccessControl` del recurso `AWS::S3::Bucket` que contiene el módulo. A continuación se muestra el fragmento de plantilla para `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
```

A continuación, el módulo anterior se anida dentro de un módulo principal, `My::S3::SampleBucket::MODULE`. El módulo principal, `My::S3::SampleBucket::MODULE`, establece los parámetros del módulo secundario de las siguientes maneras:

- Establece el parámetro `AccessControl` de `My::S3::SampleBucketPrivate::MODULE` a `Private`.
- Para `BucketName`, define un parámetro de módulo, que permitirá que el nombre del bucket se especifique en la plantilla (o módulo) que contiene `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
```

Especificación de restricciones para parámetros de módulo

Los parámetros del módulo no admiten la restricción de ejecución. Para comprobar la restricción en un parámetro de módulo, cree un parámetro de plantilla con las restricciones deseadas. A continuación, haga referencia a ese parámetro de plantilla en el parámetro del módulo. Para obtener más información sobre cómo definir parámetros de plantilla, consulte [Sintaxis de Parameters de la plantilla de CloudFormation](#).

Recursos del módulo de referencia en las plantillas de CloudFormation

En las plantillas de CloudFormation, a menudo, es necesario establecer las propiedades de un recurso en función del nombre o la propiedad de otro recurso. Para obtener más información, consulte [Recursos de referencia](#).

Para hacer referencia a un recurso contenido en un módulo de la plantilla de CloudFormation, debe combinar dos nombres lógicos:

- El nombre lógico que asignó al módulo en sí cuando lo incluyó en la plantilla.
- El nombre lógico del recurso específico de ese módulo.

Puede combinar estos dos nombres lógicos con o sin usar un punto (.) entre ellos. Por ejemplo, si el nombre lógico del módulo es `MyModule` y el nombre lógico del recurso es `MyBucket`, puede hacer referencia a ese recurso como `MyModule.MyBucket` o `MyModuleMyBucket`.

Para encontrar los nombres lógicos de los recursos de un módulo, puede consultar el esquema del módulo, que está disponible en el registro de CloudFormation o mediante la operación [DescribeType](#). El esquema muestra todos los recursos y sus nombres lógicos que forman parte del módulo.

Una vez que tenga el nombre lógico completo, puede usar las funciones de CloudFormation, como `GetAtt` y `Ref` para acceder a los valores de propiedad en los recursos del módulo.

Por ejemplo, tiene un módulo `My::S3::SampleBucket::MODULE` que contiene un recurso `AWS::S3::Bucket` con el nombre lógico `S3Bucket`. Para hacer referencia al nombre de este bucket mediante la función `Ref`, combine el nombre del módulo de la plantilla (`MyBucket`) con el nombre lógico del recurso del módulo (`S3Bucket`). El nombre lógico completo es `MyBucket.S3Bucket` o `MyBucketS3Bucket`.

Plantilla de ejemplo

La siguiente plantilla de ejemplo crea un bucket de S3 mediante el módulo `My::S3::SampleBucket::MODULE`. También crea una cola de Amazon SQS y establece su nombre para que sea el mismo que el nombre del bucket del módulo. Además, la plantilla devuelve el nombre de recurso de Amazon (ARN) del bucket de S3 creado.

```
# 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
```

Administrar los recursos de AWS como una sola unidad con pilas CloudFormation

Una pila es una colección de recursos de AWS, que puede administrar como una única unidad. En otras palabras, puede crear, actualizar y eliminar una colección de recursos mediante la creación, actualización o eliminación de pilas.

La creación de una pila implica la implementación de una plantilla de CloudFormation que especifique los recursos y sus configuraciones, que CloudFormation luego aprovisiona y configura.

La actualización de una pila implica realizar cambios en la plantilla o en los parámetros. CloudFormation compara los cambios que envíe con el estado actual de la pila y actualiza solo los recursos modificados. Al actualizar una pila, CloudFormation podría interrumpir los recursos o sustituir los recursos actualizados, en función de las propiedades que actualice. Para obtener más información acerca de las conductas de actualización, consulte [Comprensión de los comportamientos de actualización de los recursos de la pila](#).

CloudFormation ofrece dos métodos para actualizar las pilas:

- **Conjuntos de cambios:** con los conjuntos de cambios, puede obtener una vista previa de los cambios que hará CloudFormation a la pila y, a continuación, decidir si desea aplicarlos. Los conjuntos de cambios son documentos con formato JSON que resumen los cambios que hará CloudFormation a una pila. Utilice conjuntos de cambios cuando desee asegurarse de que CloudFormation no realiza cambios involuntarios o cuando desee considerar varias opciones. Por ejemplo, puede utilizar un conjunto de cambios para verificar que CloudFormation no sustituirá las instancias de la base de datos de la pila durante una actualización.
- **Actualización directa:** al actualizar directamente una pila, usted envía los cambios y CloudFormation los implementa inmediatamente. Utilice las actualizaciones directas cuando desee implementar rápidamente sus actualizaciones.

Al eliminar una pila, se eliminan los recursos asociados a ella. Una pila, por ejemplo, puede incluir todos los recursos necesarios para ejecutar una aplicación web, como, por ejemplo, un servidor web, una base de datos y reglas de red. Si ya no necesita esa aplicación web, tan solo tiene que eliminar la pila y se eliminarán todos los recursos relacionados.

 Note

Se le cobrará por los recursos de la pila por el tiempo durante el cual han estado funcionando (incluso si eliminó la pila de inmediato).

CloudFormation garantiza que todos los recursos de la pila se creen o eliminen según corresponda. Dado que CloudFormation trata los recursos de la pila como una unidad, estos deben crearse o eliminarse correctamente para poder crear o eliminar la pila. Si no es posible crear un recurso, CloudFormation restaura la pila y elimina en forma automática cualquier recurso que se haya creado. Si no es posible eliminar un recurso, los recursos restantes se retienen hasta que se pueda eliminar correctamente la pila.

Temas

- [Interfaces para gestionar las pilas](#)
- [Creación de una pila en la consola de CloudFormation](#)
- [Visualización de la información de la pila desde la consola de CloudFormation](#)
- [Actualizar la plantilla de pila](#)
- [Comprensión de los comportamientos de actualización de los recursos de la pila](#)
- [Actualización de pilas de CloudFormation con conjuntos de cambios](#)
- [Validación de las implementaciones de pilas](#)
- [Actualizar pilas directamente](#)
- [Cancelar una actualización de pila](#)
- [Eliminar una pila de la consola de CloudFormation](#)
- [Monitorizar el progreso de la pila](#)
- [Revierta su pila de CloudFormation en caso de violación de alarmas con activadores de reversión](#)
- [Detectar cambios de configuración no administrados en pilas y recursos con detección de derivación](#)
- [Cómo importar recursos de AWS a una pila de CloudFormation](#)
- [Refactorización de pilas](#)
- [Tipo de recurso compatible](#)
- [Uso de los enlaces de creación rápida para crear pilas de CloudFormation](#)
- [Ejemplos de comandos de operación de pila de CloudFormation para la AWS CLI y PowerShell](#)

Interfaces para gestionar las pilas

Puede administrar las pilas de CloudFormation con cualquiera de las siguientes interfaces:

- **Consola de CloudFormation:** proporciona una interfaz web que se puede utilizar para obtener acceso a sus pilas. Puede acceder a la consola de CloudFormation si inicia sesión en la Consola de administración de AWS, utiliza el cuadro de búsqueda de la barra de navegación para buscar CloudFormation y, a continuación, elige CloudFormation en los resultados de la búsqueda.
- **AWS Command Line Interface** proporciona comandos para numerosos servicios de AWS, incluido CloudFormation, y es compatible con Windows, Mac y Linux. Para obtener información sobre los comandos de CloudFormation, consulte [cloudformation](#) en la Referencia de comandos de AWS CLI.
- **Herramientas de AWS para PowerShell:** un conjunto de módulos de PowerShell basados en la funcionalidad expuesta en SDK para .NET. Las herramientas para PowerShell le permiten llevar a cabo operaciones mediante script en sus recursos de AWS desde la línea de comandos de PowerShell. Puede encontrar los cmdlets para CloudFormation en la [Referencia de cmdlets de Herramientas de AWS para PowerShell](#).
- **API de consulta:** proporciona acciones de API de nivel bajo a las que se llama mediante solicitudes HTTPS. Si realiza llamadas a la API en su aplicación, debe escribir el código para gestionar los detalles de bajo nivel, como, por ejemplo, la generación del hash para firmar la solicitud. Para obtener más información sobre las acciones de la API para CloudFormation, consulte [Acciones](#) en la AWS CloudFormation Referencia de la API de .
- **AWS SDK:** proporcionan API específicas de cada lenguaje y se encargan de muchos de los detalles de conexión, tales como, el cálculo de firmas, el control de reintentos de solicitudes y el control de errores. Para obtener más información, consulte [Herramientas para crear en AWS](#).
- **AWS Cloud Development Kit (AWS CDK):** AWS CDK es un marco de desarrollo de software de código abierto que le permite definir la infraestructura de AWS mediante lenguajes de programación conocidos como TypeScript, Python, Java y .NET. Con CDK, puede modelar los recursos de aplicación y, a continuación, aprovisionarlos mediante CloudFormation directamente desde su entorno de desarrollo integrado (IDE). Para obtener más información, consulte [AWS Cloud Development Kit \(AWS CDK\)](#).

Creación de una pila en la consola de CloudFormation

Puede crear una plantilla de pila y, a continuación, utilizarla para crear una pila mediante la consola de CloudFormation o una herramienta de línea de comandos. En la consola se proporciona una interfaz guiada por un asistente con opciones predefinidas que agiliza el proceso de creación de la pila.

Temas

- [Creación de una pila](#)
- [Configurar las opciones la pila](#)
- [Obtenga una vista previa de la configuración de la pila](#)

Creación de una pila

Siga los pasos descritos en esta sección para implementar su plantilla y crear una pila.

Requisitos previos

- Tiene que haber creado una plantilla de pila. Para obtener más información, consulte [Trabajo con plantillas de CloudFormation](#).

Creación de una pila (consola)

1. Abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation>.
2. En la barra de navegación de la parte superior de la pantalla, elija la Región de AWS en la que creó la pila.
3. En la página Pilas, elija Crear pila en la parte superior derecha y, a continuación, elija Con recursos nuevos (estándar).

Como alternativa, puede elegir la opción Con recursos existentes (importar recursos) para importar recursos de AWS existentes que estén descritos en la plantilla. Para obtener más información acerca de esta opción, consulte [Cómo importar recursos de AWS a una pila de CloudFormation](#).

4. En la página Crear pila, proceda de uno de los modos siguientes:

- Para usar una plantilla existente, en Requisito previo: preparar la plantilla, elija Seleccione una plantilla existente. A continuación, en Especificar plantilla, seleccione URL de Amazon S3 o Cargar un archivo de plantilla en función de la ubicación de la plantilla.
- Si elige URL de Amazon S3, proporcione una URL al archivo de plantilla en un bucket de S3.

Si la plantilla incluye pilas anidadas (p. ej., pilas descritas en otros documentos de plantilla ubicados en subdirectorios), asegúrese de que su bucket de S3 contiene los archivos y directorios necesarios.

Si tiene una plantilla en un bucket compatible con el control de versiones, puede especificar una versión concreta de la plantilla al adjuntar `?versionId=version-id` a la URL: Para obtener más información sobre bucket con control de versiones, consulte [Administración de objetos en un bucket con control de versiones habilitado](#) en la Guía del usuario de la consola de Amazon Simple Storage Service.

La URL debe dirigir a una plantilla con un tamaño máximo de 1 MB que esté almacenada en un bucket de S3 en el que tengas permisos de lectura. La URL puede tener una longitud máxima de 1024 caracteres. Algunos recursos pueden requerir que el bucket esté en la misma región que la pila.

- Si elige Cargar un archivo de plantilla, seleccione Elegir archivo para elegir un archivo de plantilla de su computadora local. El tamaño del archivo de plantilla debe ser de 1 MB o menos.

Una vez que haya elegido la plantilla, CloudFormation carga el archivo y muestra la URL de S3. CloudFormation lo carga en un bucket de Amazon S3 de su Cuenta de AWS. Si ya tiene un bucket de S3 creado por CloudFormation en su Cuenta de AWS, CloudFormation agrega la plantilla a ese bucket. Si todavía no tiene un bucket existente creado con CloudFormation, crea un bucket único para cada región en la que carga un archivo de plantilla.

A continuación, se muestran las consideraciones que existen cuando se utilizan buckets de S3 creados por CloudFormation:

- Cualquiera con permisos de Amazon S3 en su Cuenta de AWS puede acceder a los buckets.
- CloudFormation crea los buckets con el cifrado del lado del servidor habilitado de forma predeterminada; por lo tanto, cifra todos los objetos almacenados en el bucket.

Puede administrar directamente las opciones de cifrado para los buckets creados con CloudFormation; por ejemplo, mediante la consola de Amazon S3 en <https://console.aws.amazon.com/s3/> o la AWS CLI. Para obtener más información, consulte [Establecer el comportamiento del cifrado predeterminado del lado del servidor para los buckets de Amazon S3](#), en la Guía del usuario de Amazon Simple Storage Service.

- Puede utilizar su propio bucket y administrar sus permisos cargando manualmente las plantillas a Amazon S3. Al crear o actualizar una pila, especifique la URL de Amazon S3 de un archivo de plantilla.
- Si no tiene ninguna plantilla lista, puede seleccionar Cree a partir de Infrastructure Composer para crear una plantilla con Infrastructure Composer. Para obtener más información, consulte [Infrastructure Composer](#).

5. Seleccione Siguiente para continuar y validar la plantilla.

Antes de continuar, CloudFormation valida su plantilla para identificar errores de sintaxis y algunos de semántica, como las dependencias circulares. Durante la validación, CloudFormation comprueba primero si la plantilla es un JSON válido. Si no lo es, CloudFormation comprueba si la plantilla es un YAML válido. Si ambas comprobaciones fallan, CloudFormation devuelve un error de validación de plantilla.

6. En la página Specify stack (Especificar pila), escriba un nombre de pila en el cuadro Stack name (Nombre de pila).

El nombre de pila es un identificador que le ayuda a encontrar una en particular en una lista de pilas. El nombre de una pila puede contener únicamente caracteres alfanuméricos (con distinción entre mayúsculas y minúsculas) y guiones. Debe comenzar con un carácter alfabético y no puede tener más de 128 caracteres.

7. En la sección Parámetros, especifique los valores para los parámetros que se definen en la plantilla.
8. Seleccione Siguiente para seguir creando la pila.
9. (Opcional) En la página Configurar las opciones la de pila, agregue etiquetas y otras opciones de pila. Para obtener más información, consulte [Configurar las opciones la pila](#).
10. Si la plantilla contiene recursos de IAM, en Capacidades elija Confirmando que esta plantilla pueda crear recursos de IAM para especificar que desea utilizar recursos de IAM en la plantilla. Para obtener más información, consulte [Reconocimiento de recursos de IAM en plantillas de CloudFormation](#).
11. Elija Siguiente para continuar.

12. En la página Revisar y crear, revise los detalles de la pila.

Para cambiar cualquiera de los valores antes de lanzar la pila, elija Atrás en la sección que contiene la configuración que desea cambiar.

13. (Opcional) Puede crear un conjunto de cambios para obtener una vista previa de la configuración de la pila antes de crearla. En la página Revisar y crear, seleccione Crear conjunto de cambios y siga las instrucciones. Para obtener más información, consulte [Obtenga una vista previa de la configuración de la pila](#).

14. Seleccione Enviar para lanzar su stack.

CloudFormation procederá entonces con la creación de todos los recursos definidos en la plantilla.

Puede supervisar el progreso y el estado de la creación de la pila en la pestaña Eventos de la nueva pila. Para obtener más información, consulte [Monitorizar el progreso de la pila](#).

Creación de una pila con la línea de comandos

Puede utilizar uno de los siguientes comandos:

- [create-stack](#) (AWS CLI)
- [New-CFNStack](#) (AWS Tools for Windows PowerShell)

Para ver ejemplos del uso de la línea de comandos para crear una pila, consulte [Ejemplos de comandos de operación de pila de CloudFormation para la AWS CLI y PowerShell](#).

Configurar las opciones la pila

En la página Configurar opciones de pila, puede configurar opciones para las pilas de CloudFormation como etiquetas, notificación de eventos de pila o una política de pilas.

Puede definir las siguientes opciones de pila:

Etiquetas de

Puede agregar hasta 50 pares de claves y etiquetas a su pila y a cualquier recurso para el que CloudFormation admite el etiquetado. Una etiqueta es una clave y un valor definidos por el cliente que se puede asignar a los recursos de AWS para fines como el seguimiento de los costos.

Una clave puede contener cualquier carácter alfanumérico o espacios. Las claves de etiqueta pueden tener hasta 127 caracteres.

Un valor puede contener cualquier carácter alfanumérico o espacios. Los valores de etiqueta pueden tener hasta 255 caracteres.

Después de crear, añadir, actualizar o eliminar etiquetas a nivel de pila se iniciará una actualización en la pila. Se actualizarán todos los recursos que admiten la propagación de etiquetas a nivel de pila en consecuencia.

Permisos

Un rol de servicio de IAM existente que CloudFormation puede asumir. En lugar de utilizar las credenciales de su cuenta, CloudFormation utiliza las credenciales del rol para crear la pila. Para obtener más información, consulte [Rol de servicio CloudFormation](#).

Opciones de error de pila

Especifica las opciones de error de aprovisionamiento para todas las implementaciones de pilas y operaciones de conjuntos de cambios. Para obtener más información, consulte [Elija cómo gestionar los errores al aprovisionar recursos](#).

La opción Roll back all stack resources (Restaurar todos los recursos de la pila) restaurará todos los recursos especificados en la plantilla cuando el estado de la pila sea `CREATE_FAILED` o `UPDATE_FAILED`.

Para crear operaciones, la opción Preserve successfully provisioned resources (Conservar los recursos aprovisionados correctamente) conserva el estado de los recursos correctos, mientras que los recursos con errores permanecerán en un estado con error hasta que se lleve a cabo la siguiente operación de actualización.

Para las operaciones de actualización y de conjuntos de cambios, la opción Preserve successfully provisioned resources (Conservar los recursos aprovisionados correctamente) conserva el estado de los recursos correctos y, al mismo tiempo, restaura los recursos con errores al último estado estable conocido. El estado de los recursos con error será `UPDATE_FAILED`. Los recursos sin un último estado estable conocido se eliminarán en la siguiente operación de pila.

También puede definir las siguientes opciones avanzadas para la creación de la pila:

Stack policy (Política de pila)

Define los recursos que desea proteger de actualizaciones involuntarias durante una actualización de pila. De forma predeterminada, se pueden actualizar todos los recursos durante una actualización de pila.

Puede introducir la política de pilas directamente como JSON o cargar un archivo JSON que contenga la política de pilas. Para obtener más información, consulte [Impida actualizaciones en los recursos de la pila](#).

Rollback configuration (Configuración de restauración)

Permite que CloudFormation monitorice el estado de su pila durante la creación y la actualización de la pila y restaure dicha operación si la pila sobrepasa el umbral de las alarmas especificadas. Especifique las alarmas de CloudWatch que CloudFormation debe supervisar. Si alguna de las alarmas pasa al estado ALARM durante la operación de la pila o el periodo de monitoreo, CloudFormation revierte toda la operación de la pila. Para obtener más información, consulte [Revierta su pila de CloudFormation en caso de violación de alarmas con activadores de reversión](#).

Opciones de notificación

Puede especificar un tema de Amazon Simple Notification Service nuevo o ya existente adonde se envían las notificaciones sobre eventos de pila.

Si crea un tema de Amazon SNS, debe especificar un nombre y una dirección de correo electrónico, adonde se envían las notificaciones de eventos de pila.

Stack creation options (Opciones de creación de pila)

Las siguientes opciones está incluidas para la creación de la pila, pero no están disponibles como parte de las actualizaciones de la pila.

Timeout (Tiempo de espera)

Especifica la cantidad de tiempo, en minutos, que CloudFormation debe adjudicar antes de que se agote el tiempo de las operaciones de creación de pilas. Si CloudFormation no puede crear toda la pila en el tiempo asignado, se produce un error de creación porque el tiempo de espera se agota y la pila vuelve a su estado anterior.

De forma predeterminada, no hay tiempo de espera para la creación de pilas. Sin embargo, puede que determinados recursos individuales tengan sus propios tiempos de espera en función de la naturaleza del servicio que implementan. Por ejemplo, si se agota el tiempo de espera de un recurso individual de la pila, también se agota el tiempo de espera de creación de la pila aunque todavía no se haya agotado el tiempo de espera especificado para la creación de la pila.

Protección de finalización

Evita que una pila se elimine accidentalmente. Si un usuario intenta eliminar una pila con la protección de terminación habilitada, la eliminación fallará y la pila junto con su estado no cambiarán. Para obtener más información, consulte [Protección de las pilas de CloudFormation para evitar la eliminación](#).

La protección de terminación está Disabled (Deshabilitada) de forma predeterminada.

Obtenga una vista previa de la configuración de la pila

Para obtener una vista previa de cómo se configurará una pila de CloudFormation antes de crear la pila, cree un conjunto de cambios. Esta funcionalidad le permite examinar diversas configuraciones y realizar correcciones y cambios en la pila antes de ejecutar el conjunto de cambios. Para obtener más información acerca de conjuntos de cambios, consulte [Actualización de pilas de CloudFormation con conjuntos de cambios](#).

Creación de un conjunto de cambios para una nueva pila

Para crear un conjunto de cambios para una pila nueva, seleccione la plantilla de la pila y especifique la configuración de la pila como lo haría si estuviera creando una nueva pila. A continuación, seleccione crear un nuevo conjunto de cambios en lugar de una pila nueva.

Creación de un conjunto de cambios para una nueva pila

1. En la página Revisar y crear, seleccione Crear conjunto de cambios.
2. En el cuadro de diálogo Crear conjunto de cambios, ingrese el nombre del conjunto de cambios y una descripción, si lo desea. Elija Create change set (Crear conjunto de cambios).

Cuando se crea un conjunto de cambios para una nueva pila, CloudFormation hace lo siguiente:

- Lanza una nueva pila con un estado de REVIEW_IN_PROGRESS.
- Crea un conjunto de cambios para la pila nueva que refleja la configuración de la pila que ha especificado en los pasos anteriores.

CloudFormation muestra la página Change sets (Conjuntos de cambios) para la pila. Mientras CloudFormation crea el conjunto de cambios, su estado es CREATE_IN_PROGRESS y el estado de ejecución es UNAVAILABLE. Cuando CloudFormation lleva a cabo correctamente la creación del conjunto de cambios, establece el estado como CREATE_COMPLETE y su estado de

ejecución es AVAILABLE. El estado de la pila se actualiza a REVIEW_IN_PROGRESS. En este momento, puede ejecutar el conjunto de cambios para completar la creación de la nueva pila.

En la sección Changes (Cambios), CloudFormation muestra la configuración propuesta de su pila.

Si CloudFormation no logra crear el conjunto de cambios, establece el estado como CREATE_FAILED. Corrija el error que se muestra en el campo Status reason (Motivo del estado) y, a continuación, cree un nuevo conjunto de cambios. En esta fase, puede probar diversas configuraciones y realizar correcciones y cambios en la pila antes de ejecutar el siguiente conjunto de cambios.

3. Para completar la creación de una nueva pila basada en el conjunto de cambios, elija Execute (Ejecutar) y, a continuación, elija Execute change set (Ejecutar conjunto de cambios).

Visualización de la información de la pila desde la consola de CloudFormation

Una vez que haya creado una pila de CloudFormation, puede utilizar la Consola de administración de AWS para ver sus datos y recursos.

Para ver la información de su pila de CloudFormation

1. Inicie sesión en la Consola de administración de AWS y abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation>.
2. En la barra de navegación de la parte superior de la pantalla, elija la Región de AWS en donde se encuentra la pila.
3. En la página Pilas, seleccione el nombre de la pila. CloudFormation muestra la información de la pila que haya seleccionado.

Note

Si la pila se ha eliminado, puede encontrarla mediante la opción de Filtrar estado. Para obtener más información, consulte [Visualización de las pilas eliminadas desde la consola de CloudFormation](#).

Puede ver la siguiente información de la pila:

Stack info (Info de la pila)

Muestra información general sobre la pila y su configuración, como:

Descripción general

Muestra el nombre de la pila, el ID de pila, la pila raíz y el [rol de IAM](#), junto con información de estado, como estado de la pila, estado de desviación y protección de terminación.

Etiquetas

Muestra cualquier etiqueta que esté asociada a la pila.

Stack policy (Política de pila)

Describe los recursos de la pila protegidos frente a las actualizaciones de la pila. Para que pueda actualizar estos recursos, deben permitirse explícitamente durante la actualización de una pila. Para obtener más información, consulte [Impida actualizaciones en los recursos de la pila](#).

Rollback configuration (Configuración de restauración)

Muestra las alarmas de CloudWatch que ha especificado para que CloudFormation monitoree durante la operación de la pila o el periodo de monitorización especificado. Si alguna de las alarmas pasa al estado ALARM durante la operación de la pila o el periodo de monitoreo, CloudFormation revierte toda la operación de la pila. Para obtener más información, consulte [Revierta su pila de CloudFormation en caso de violación de alarmas con activadores de reversión](#).

Opciones de notificación

Muestra el tema de Amazon Simple Notification Service adonde se envían las notificaciones sobre eventos de pila, si se especifica.

Events

Muestra las operaciones que se siguen al crear, actualizar o eliminar la pila. Para obtener más información, consulte [Monitorizar el progreso de la pila](#).

A todos los eventos activados por una determinada operación de la pila se les asigna el mismo token de solicitud de cliente, que puede utilizar para hacer un seguimiento de las operaciones. Las operaciones de la pila que se inician desde la consola utilizan el formato de token Console-OperaciónDePila-ID, lo que le permite identificar fácilmente la operación de la pila. Por ejemplo,

si crea una pila mediante la consola, a cada evento de pila resultante se le asignará el mismo token en el siguiente formato: Console-CreateStack-7f59c3cf-00d2-40c7-b2ff-e75db0987002.

Recursos

Muestra los recursos que forman parte de la pila.

Salidas

Muestra salidas declaradas en la plantilla de la pila. Para obtener más información, consulte [Obtención de salidas exportadas de una pila implementada de CloudFormation](#).

Parameters

Muestra los parámetros de la pila y sus valores.

En las pilas que contienen parámetros de Systems Manager, la columna Valor resuelto muestra los valores que se utilizan en la definición de pila para los parámetros de Systems Manager. Para obtener más información, consulte [Definición de recursos existentes en tiempo de ejecución con tipos de parámetros proporcionados por CloudFormation](#).

Plantilla

Muestra la plantilla de la pila.

En las pilas que contienen macros, elija View original template (Ver plantilla original) para ver la plantilla enviada por el usuario o View processed template (Ver plantilla procesada) a fin de ver la plantilla después de que CloudFormation haya procesado las macros a las que se hace referencia. CloudFormation usa la plantilla procesada para crear o actualizar su pila.

Conjuntos de cambios

Se muestran los conjuntos de cambios de la pila.

Para obtener más información, consulte [Visualización de un conjunto de cambios para una pila de CloudFormation](#).

Sincronización de Git

Se muestra el panel de sincronización de Git de la pila.

Para obtener más información, consulte [Panel de estado de sincronización de Git](#).

Actualizar la plantilla de pila

Para modificar los recursos o las propiedades de una pila de CloudFormation, debe actualizar la plantilla de la pila. Comience con la plantilla existente para esa pila y realice los cambios en ella. Si tiene su plantilla almacenada en un sistema de control de recursos, utilice una copia de dicha plantilla como punto de partida. De lo contrario, puede obtener una copia de la plantilla desde CloudFormation.

Si solo desea modificar los parámetros o los ajustes de una pila (como por ejemplo el tema de Amazon SNS de una pila), puede volver a utilizar la plantilla de la pila existente sin obtener una copia.

Puede actualizar una plantilla de pila de CloudFormation mediante un editor de texto o [Infrastructure Composer](#).

Actualización de una plantilla de pila existente mediante Infrastructure Composer

1. Inicie sesión en la Consola de administración de AWS y abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation>.
2. En la página Pilas, seleccione el nombre de la pila que desea actualizar.
3. Seleccione la pestaña Plantilla y, a continuación, elija Ver en Infrastructure Composer.

CloudFormation abre la plantilla en Infrastructure Composer.

4. Utilice uno de los siguientes métodos para actualizar la plantilla:
 - Interfaz de lienzo: aquí puede arrastrar y soltar desde la paleta Recursos. Para configurar los recursos, haga doble clic en una tarjeta para abrir el panel Propiedades del recurso. Conecte los recursos según sea necesario. Para obtener instrucciones detalladas sobre el uso de la interfaz de lienzo, consulte [How to compose in AWS Infrastructure Composer](#).
 - Interfaz de plantilla: cambie de la interfaz de lienzo a la interfaz de plantilla. Realice actualizaciones en línea del código de la plantilla. Cambie entre los formatos JSON y YAML según sea necesario.
5. Elija Validar para comprobar si hay errores de sintaxis en la plantilla.
6. Cuando tenga todo listo para exportar los cambios a CloudFormation, elija Actualizar plantilla.

Para actualizar una plantilla de conjunto de pilas con la AWS CLI

1. Para obtener la plantilla de la pila que desea actualizar, utilice el comando [get-template](#) de la CLI.
2. Copie la plantilla, péguela en un archivo de texto, modifíquela y guárdela. Copie únicamente la plantilla. El comando encierra la plantilla entre comillas, pero no copia las comillas alrededor de la plantilla. La plantilla comienza con una llave de apertura y termina con la llave de cierre final. Especifique los cambios a los recursos de la pila en este archivo.

Tenga en cuenta los siguientes puntos cuando realice cambios en la plantilla:

- No puede agregar, modificar o eliminar un parámetro utilizado por un recurso que no admite actualizaciones.
- Para la mayoría de los recursos, cambiar el nombre lógico de un recurso equivale a la eliminación de ese recurso y a la sustitución por otro nuevo. Cualquier otro recurso que dependa del recurso cuyo nombre se ha cambiado también tendrá que actualizarse y podría provocar su reemplazo. Otros recursos requieren que actualice una propiedad (no solo el nombre lógico) para iniciar una actualización.
- Algunos recursos pueden tener restricciones en cuanto a los valores que se pueden establecer para determinadas propiedades. Por ejemplo, los cambios en la propiedad `AllocatedStorage` de una instancia de base de datos de RDS deben ser superiores al valor actual. Si la actualización infringe estas reglas, esa parte fallará.
- La actualización de un recurso también puede afectar a otros que hagan referencia a él. Si usa funciones como [La función de Ref](#) o [La función de Fn::GetAtt](#) para establecer una propiedad basada en otro recurso, CloudFormation también actualizará el recurso de referencia cuando cambie el recurso al que se hace referencia.
- Consulte [Referencia de tipos de recursos y propiedades de AWS](#) para obtener información sobre los efectos de actualizar propiedades específicas de recursos. Para cada propiedad, los efectos de una actualización serán uno de los siguientes:
 - Requisitos de actualización: [sin interrupción](#)
 - Requisitos de actualización: [alguna interrupción](#)
 - Requisitos de actualización: [reemplazo](#)
- Puede comprobar la sintaxis JSON o YAML de la plantilla con el comando [validate-template](#) de la CLI o al especificar la plantilla en la consola. La consola realiza la validación automáticamente. Sin embargo, estos métodos solo comprueban la sintaxis de la plantilla y no validan que los valores de

propiedad que ha especificado para un recurso sean válidos para ese recurso. Para validaciones más complejas o para comprobar las prácticas recomendadas, también puede utilizar herramientas adicionales como [CloudFormation Linter \(cfn-lint\)](#) y [CloudFormation Rain \(rain fmt\)](#).

Note

A veces, CloudFormation no permite ciertos cambios que intente realizar y le dirá que el cambio no está permitido. Este mensaje puede ocurrir de manera asincrónica, ya que CloudFormation crea y actualiza recursos en un orden no determinista de forma predeterminada.

Comprensión de los comportamientos de actualización de los recursos de la pila

Al enviar una actualización, CloudFormation actualiza recursos en función de las diferencias entre lo que envía y la plantilla actual de la pila. Los recursos que no han cambiado se ejecutan sin interrupción durante el proceso de actualización. Para recursos actualizados, CloudFormation utiliza uno de los siguientes comportamientos de actualización:

Actualización sin interrupciones

CloudFormation actualiza el recurso sin interrumpir el funcionamiento de dicho recurso y sin cambiar el ID físico del recurso. Por ejemplo, si actualiza cierta propiedad en un recurso [AWS::CloudTrail::Trail](#), CloudFormation actualiza el registro de seguimiento sin interrupción.

Actualizaciones con algunas interrupciones

CloudFormation actualiza el recurso con algunas interrupciones. Por ejemplo, si actualiza cualquier propiedad en un recurso [AWS::EC2::Instance](#), podrían producirse algunas interrupciones en la instancia mientras CloudFormation y Amazon EC2 vuelven a configurar la instancia.

Reemplazo

CloudFormation vuelve a crear el recurso durante una actualización, lo que también genera un nuevo ID físico. Por lo general, CloudFormation crea en primer lugar el recurso de sustitución, cambia referencias de otros recursos dependientes para que apunten al recurso de sustitución y, a continuación, elimina el recurso antiguo. Por ejemplo, si actualiza la propiedad

`AvailabilityZone` de un tipo de recurso [AWS::EC2::Instance](#), CloudFormation crea un recurso nuevo y sustituye el recurso de la instancia de EC2 por el nuevo.

Si agregas o eliminas una propiedad que requiere un reemplazo, también se activará una actualización. Esta actualización se llevará a cabo incluso si el valor real de la propiedad no cambia.

El método que CloudFormation utiliza depende de la propiedad que actualice para un tipo de recurso determinado. El comportamiento de actualización para cada propiedad se describe en la [referencia de tipos de recursos y propiedades de AWS](#).

En función del comportamiento de actualización, puede decidir cuándo modificar los recursos para reducir el impacto de estos cambios en su aplicación. En particular, puede planificar cuándo deben sustituirse los recursos durante una actualización. Por ejemplo, si actualiza la propiedad `Port` de un tipo de recurso [AWS::RDS::DBInstance](#), CloudFormation sustituye la instancia de base de datos creando una nueva instancia de base de datos con la configuración del puerto actualizada y elimina la instancia de base de datos antigua. Antes de la actualización, podría pensar en hacer lo siguiente para prepararse para la sustitución de la base de datos:

- Realice una instantánea de las actuales bases de datos.
- Prepare una estrategia para decidir cómo tratarán una interrupción las aplicaciones que utilizan la instancia de base de datos mientras se sustituye la instancia de base de datos.
- Asegúrese de que las aplicaciones que utilizan la instancia de base de datos consideran la configuración actualizada del puerto y cualquier otra configuración que haya realizado.
- Utilice la instantánea de base de datos para restaurar las bases de datos en la nueva instancia de base de datos.

Este ejemplo no es exhaustivo, pero tiene por objeto darle una idea de las cosas que debe planificar cuando se sustituye un recurso durante una actualización.

Note

Si la plantilla incluye una o varias [pilas anidadas](#), CloudFormation también inicia una actualización por cada pila anidada. Esto es necesario para determinar si se han modificado las pilas anidadas. CloudFormation actualiza únicamente los recursos de las pilas anidadas que tienen cambios especificados en plantillas correspondientes.

Actualización de pilas de CloudFormation con conjuntos de cambios

Cuando necesita actualizar una pila, comprender cómo afectarán los cambios a los recursos que se están ejecutando antes de implementarlos puede ayudarle a actualizar pilas con confianza. Los conjuntos de cambios ofrecen una vista previa del efecto de los cambios propuestos sobre los recursos en ejecución, lo cual incluye el efecto sobre las propiedades y los atributos de los recursos. Ya sea que los cambios eliminen o sustituyan algún recurso crítico, CloudFormation hace los cambios en la pila solo cuando se decida ejecutar el conjunto de cambios, lo que permite tomar la decisión de implementar los cambios propuestos o de explorar otros cambios mediante la creación de un conjunto nuevo. Puede crear y administrar conjuntos de cambio mediante la consola de CloudFormation, la AWS CLI o la API de CloudFormation.

Temas

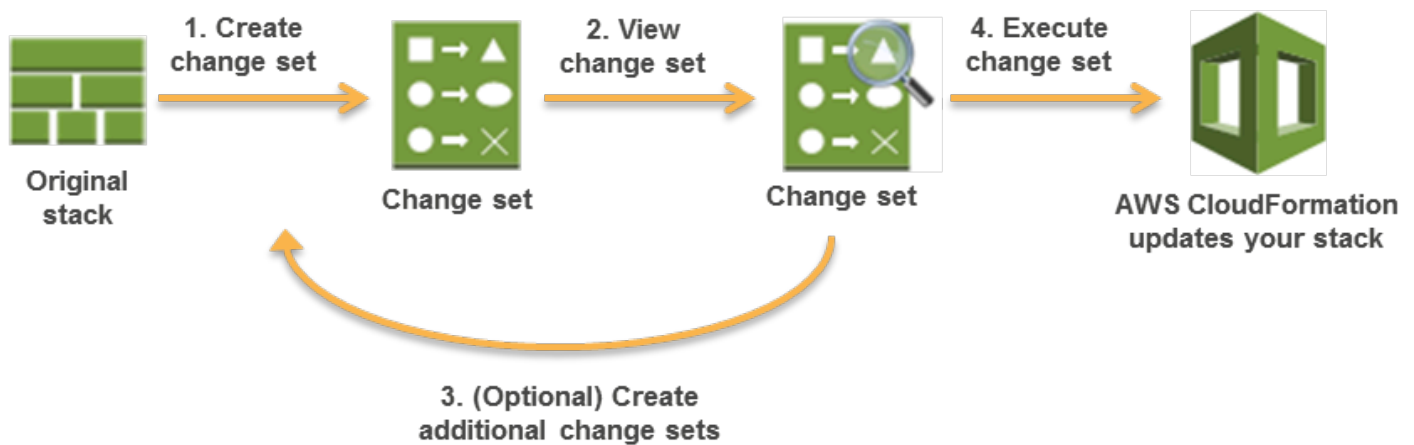
- [Creación de un conjunto de cambios para una pila de CloudFormation](#)
- [Visualización de un conjunto de cambios para una pila de CloudFormation](#)
- [Uso de conjuntos de cambios con reconocimiento de desviación](#)
- [Ejecución de un conjunto de cambios para una pila de CloudFormation](#)
- [Eliminación de un conjunto de cambios para una pila de CloudFormation](#)
- [Ejemplo de conjuntos de cambio para pilas de CloudFormation](#)
- [Conjuntos de cambios para pilas anidadas](#)

Important

Cambiar conjuntos no indica si CloudFormation va a actualizar correctamente una pila. Por ejemplo, un conjunto de cambios no comprueba si se va a sobrepasar una cuota de cuenta, si va a actualizar un recurso que no admite actualizaciones o si no tiene suficientes permisos para modificar un recurso; todo esto puede causar el error de actualización de una pila. Si una actualización falla, CloudFormation intenta devolver los recursos a su estado original.

Información general de conjunto de cambios

El siguiente diagrama resume cómo utilizar conjuntos de cambio para actualizar una pila:



1. Cree un conjunto de cambios enviando cambios para la pila que desea actualizar. Puede enviar una plantilla de pila modificada o valores de parámetros de entrada modificados. CloudFormation compara la pila con los cambios que ha enviado para generar el conjunto de cambios; no hace cambios en la pila en este punto.
2. Visualice el conjunto de cambios para ver qué ajustes y recursos de pila cambiarán. Por ejemplo, puede ver qué recursos CloudFormation agregará, modificará o eliminará. Además, puede acceder a una comparación del estado previo y posterior de las propiedades y los atributos de los recursos, como las etiquetas, que modificará CloudFormation.
3. Opcional: si desea tener en cuenta otros cambios antes de decidir qué cambios hacer, cree conjuntos de cambios adicionales. La creación de varios conjuntos de cambios ayuda a comprender y evaluar la manera en que los cambios afectarán los recursos y las propiedades. Puede crear todos los conjuntos de cambios que necesite.
4. Ejecute el conjunto de cambios que contiene los cambios que desee aplicar a la pila. CloudFormation actualiza la pila con dichos cambios.

Note

Después de ejecutar un cambio, CloudFormation elimina todos los conjuntos de cambios asociados a la pila, ya que no son aplicables a la pila actualizada.

También puede eliminar conjuntos de cambios para impedir la ejecución de un conjunto de cambios que no debería aplicarse.

Creación de un conjunto de cambios para una pila de CloudFormation

Para crear un conjunto de cambios de una pila en ejecución, envíe los cambios que quiera hacer proporcionando una plantilla modificada, nuevos valores de parámetros de entrada o ambos.

CloudFormation genera un conjunto de cambios comparando la pila con los cambios que envió.

Puede modificar una plantilla antes de crear el conjunto de cambios o durante el proceso de creación del conjunto de cambios.

Create a change set (console)

Creación de un conjunto de cambios

1. Abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation>.
2. En la barra de navegación de la parte superior de la pantalla, elija la Región de AWS.
3. En la página Pilas, seleccione la pila en ejecución para la que desea crear un conjunto de cambios.
4. En el panel de detalles de la pila, seleccione Acciones de pila y, a continuación, seleccione Crear conjunto de cambios para pila actual.
5. En la página Create change set for **nombre-pila** (Crear conjunto de cambios para nombre-pila), realice una de las siguientes operaciones para modificar los valores de los parámetros de entrada, especifique la ubicación de una plantilla actualizada o modifique la plantilla:

Tarea	Acción
Modificación de los valores de los parámetros de entrada	Seleccione Utilizar plantilla actual y, a continuación, seleccione Siguiente para introducir o modificar valores de los parámetros de entrada.
Especificación de la ubicación de una plantilla actualizada	Si modificó la plantilla, seleccione Sustituir plantilla actual y, a continuación, realice alguna de las siguientes operaciones: • En el caso de una plantilla almacenada en un bucket de Amazon S3, seleccione Amazon S3 URL (URL de Amazon S3). Escriba o pegue la URL de la plantilla y luego seleccione Next (Siguiente).

Tarea	Acción
	Si tiene una plantilla en un bucket compatible con el control de versiones, puede especificar una versión concreta de la plantilla al adjuntar <code>?versionId= <i>version-id</i></code> a la URL. Para obtener más información, consulte Trabajo con objetos en un bucket con control de versiones habilitado en la Guía del usuario de la consola de Amazon Simple Storage Service. • En el caso de una plantilla almacenada de manera local en el equipo, seleccione Upload a template file (Cargar un archivo de plantilla). Seleccione Choose File (Seleccionar archivo) para navegar hasta el archivo, selecciónelo y, a continuación, seleccione Next (Siguiente).

Modificación de la plantilla

Si no modificó la plantilla, seleccione Editar plantilla en Infrastructure Composer y, a continuación, seleccione Editar en Infrastructure Composer. Se le redirigirá al AWS Infrastructure Composer. Una vez que modificó la plantilla, seleccione Crear conjunto de cambios, luego Confirmar y volver a CloudFormation para volver a la página Crear un conjunto de cambios para *nombre-pila* y, a continuación, seleccione Siguiente.

6. En la página Especificar detalles de la pila, especifique el nombre de la pila para el conjunto de cambios y, de manera opcional, agregue una descripción del conjunto para identificar su propósito en la sección Información general. Si la plantilla incluye parámetros, en la página Specify stack details (Especificar detalles de la pila), ingrese o modifique los valores de parámetros de entrada pertinentes y, a continuación, seleccione Next (Siguiente).

Si reutiliza la plantilla de la pila, CloudFormation rellenará cada parámetro con el valor actual en la pila, con la excepción de los parámetros declarados con el atributo NoEcho. Para utilizar los valores existentes para esos parámetros, seleccione Use existing value (Usar valor existente).

Para obtener más información sobre el uso de NoEcho para enmascarar información confidencial y el uso de parámetros dinámicos para la administración de secretos, consulte la práctica recomendada de [No integre credenciales en sus plantillas](#).

7. En la página Configurar opciones de pila, actualice las etiquetas de la pila, el rol de servicio de IAM, la política de pila, la configuración de reversión, el tema de notificación de Amazon SNS (si corresponde) o cambie los conjuntos.

 Note

Los conjuntos de cambio para las pilas anidadas están, de manera predeterminada, Habilitadas, lo cual crea conjuntos de cambio para todas las pilas anidadas especificadas en la plantilla. Para crear un conjunto de cambios solo para la pila actual, seleccione Disabled (Desabilitado). Para obtener más información sobre los conjuntos de cambios para las pilas anidadas, consulte [Conjuntos de cambios para pilas anidadas](#).

8. Si la plantilla incluye recursos de IAM, en Capacidades seleccione Acepto que CloudFormation pueda crear recursos de IAM. Los recursos de IAM pueden modificar los permisos de su cuenta de AWS; revise estos recursos para garantizar que solo se le permitan las acciones que planea. Para obtener más información, consulte [Reconocimiento de recursos de IAM en plantillas de CloudFormation](#).
9. Elija Siguiente para continuar.
10. En la página Review **stack-name** (Revisar nombre-pila), revise los cambios del conjunto de cambios.
11. Seleccione Submit (Enviar).

Se le redirigirá a la pestaña Changes (Cambios) de la página de detalles del conjunto de cambios. Mientras CloudFormation genera un conjunto de cambios, el estado del conjunto es CREATE_PENDING. Una vez que crea el conjunto de cambios, CloudFormation establece el estado como CREATE_COMPLETE. En la sección Changes (Cambios), CloudFormation enumera todos los cambios que realizará en la pila. Para obtener más información, consulte [Visualización de un conjunto de cambios para una pila de CloudFormation](#).

Seleccione Ver detalles en la columna Cambios a nivel de propiedad para poder ver los cambios que se realizaron en el nivel de propiedad.

Si CloudFormation no logra crear el conjunto de cambios (informa un estado FAILED), arregle el error que se muestra en el campo Status (Estado) y cree nuevamente el conjunto de cambios.

12. Después de controlar los cambios, seleccione Ejecutar conjunto de cambios.

Create a change set for nested stacks (console)

Creación de un conjunto de cambios para pilas anidadas

1. Abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation>.
2. En la barra de navegación de la parte superior de la pantalla, elija la Región de AWS.
3. En la página Pilas, seleccione la pila en ejecución para la que desea crear un conjunto de cambios.
4. En el panel de detalles de la pila, seleccione Acciones de pila y, a continuación, seleccione Crear conjunto de cambios para pila actual.
5. En la página Create change set for **nombre-pila** (Crear conjunto de cambios para nombre-pila), realice una de las siguientes operaciones para modificar los valores de los parámetros de entrada, especifique la ubicación de una plantilla actualizada o modifique la plantilla:

Tarea	Acción
Modificación de los valores de los parámetros de entrada	Seleccione Utilizar plantilla actual y, a continuación, seleccione Siguiente para introducir o modificar valores de los parámetros de entrada.
Especificación de la ubicación de una plantilla actualizada	Si modificó la plantilla, seleccione Sustituir plantilla actual y, a continuación, realice alguna de las siguientes operaciones: • En el caso de una plantilla almacenada en un bucket de Amazon S3, seleccione Amazon S3 URL (URL de Amazon S3). Escriba o pegue la URL de la plantilla y luego seleccione Next (Siguiente). Si tiene una plantilla en un bucket compatible con el control de versiones, puede especificar una versión concreta de la plantilla al adjuntar <code>?versionId= version-id</code> a la URL. Para obtener más información, consulte Trabajo con objetos en un bucket con control de versiones habilitado en la Guía del usuario de la consola de Amazon Simple Storage Service. • En el caso de una plantilla almacenada de manera local en el equipo, seleccione Upload a template file (Cargar un archivo de plantilla). Seleccione Choose File (Seleccionar archivo)

Tarea	Acción
	para navegar hasta el archivo, selecciónelo y, a continuación, seleccione Next (Siguiente).
Modificación de la plantilla	Si no modificó la plantilla, seleccione Editar plantilla en Infrastructure Composer y, a continuación, seleccione Editar en Infrastructure Composer. Se le redirigirá al AWS Infrastructure Composer. Una vez que modificó la plantilla, seleccione Crear conjunto de cambios, luego Confirmar y volver a CloudFormation para volver a la página Crear un conjunto de cambios para nombre-pila y, a continuación, seleccione Siguiente.

- En la página Especificar detalles de la pila, especifique el nombre de la pila para el conjunto de cambios y, de manera opcional, agregue una descripción del conjunto para identificar su propósito en la sección Información general. Si la plantilla incluye parámetros, en la página Specify stack details (Especificación de detalles de la pila), ingrese o modifique los valores de parámetros de entrada pertinentes y, a continuación, seleccione Next (Siguiente).

Si reutiliza la plantilla de la pila, CloudFormation rellenará cada parámetro con el valor actual en la pila, con la excepción de los parámetros declarados con el atributo NoEcho. Para utilizar los valores existentes para esos parámetros, seleccione Use existing value (Usar valor existente).

Para obtener más información sobre el uso de NoEcho para enmascarar información confidencial y el uso de los parámetros dinámicos para la administración de secretos, consulte la práctica recomendada de [No integre credenciales en sus plantillas](#).

- En la página Configurar opciones de pila, actualice las etiquetas de la pila, el rol de servicio de IAM, la política de pila, la configuración de reversión, el tema de notificación de Amazon SNS (si corresponde) o cambie los conjuntos. Para obtener más información, consulte [Configurar las opciones de la pila](#).

Note

Los conjuntos de cambio para las pilas anidadas están, de manera predeterminada, Habilitadas, lo cual crea conjuntos de cambio para todas las pilas anidadas especificadas en la plantilla. Para obtener más información sobre los conjuntos de cambios para las pilas anidadas, consulte [Conjuntos de cambios para pilas anidadas](#).

8. Si la plantilla incluye recursos de IAM, en Capacidades seleccione Acepto que CloudFormation pueda crear recursos de IAM. Los recursos de IAM pueden modificar los permisos de su cuenta de AWS; revise estos recursos para garantizar que solo se le permitan las acciones que planea. Para obtener más información, consulte [Reconocimiento de recursos de IAM en plantillas de CloudFormation](#).
9. Elija Siguiente para continuar.
10. En la página Review **stack-name** (Revisar nombre-pila), revise los cambios del conjunto de cambios.
11. Elija Enviar.

 Note

Los conjuntos de cambios a nivel de propiedad de CloudFormation no resuelven las referencias entre pilas al crear conjuntos de cambios para pilas anidadas. Los conjuntos de cambios pueden marcar los recursos de una pila secundaria para su reemplazo condicional si hacen referencia a la salida de una pila principal y esta última se ha modificado

Se le redirigirá a la pestaña Changes (Cambios) de la página de detalles del conjunto de cambios. Mientras CloudFormation genera un conjunto de cambios, el estado del conjunto es CREATE_PENDING. Una vez que crea el conjunto de cambios, CloudFormation establece el estado como CREATE_COMPLETE. En la sección Changes (Cambios), CloudFormation enumera todos los cambios que realizará en la pila. Para obtener más información, consulte [Visualización de un conjunto de cambios para una pila de CloudFormation](#).

Si CloudFormation no logra crear el conjunto de cambios (informa un estado FAILED), arregle el error que se muestra en el campo Status (Estado) y cree nuevamente el conjunto de cambios.

12. Después de controlar los cambios, seleccione Ejecutar conjunto de cambios.

Creación de un conjunto de cambios (AWS CLI)

- Utilice el comando [create-change-set](#).

Envíe los cambios como opciones de comando. Puede especificar nuevos valores de parámetros, una plantilla modificada o ambos. Por ejemplo, el comando que se muestra a

continuación crea un conjunto de cambios con el nombre `SampleChangeSet` para la pila `MyStack`. El conjunto de cambios utiliza la plantilla de la pila actual, pero con un valor diferente para el parámetro `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"
```

Visualización de un conjunto de cambios para una pila de CloudFormation

Después de crear un conjunto de cambios, puede ver los cambios propuestos antes de ejecutarlos. Puede utilizar la consola de CloudFormation, la AWS CLI o la API de CloudFormation para ver los conjuntos de cambios. La consola de proporciona un resumen de los cambios y una lista detallada de los cambios en formato JSON. Las API de AWS CLI y AWS CloudFormation devolverán una lista detallada de los cambios en formato JSON.

View a change set (console)

Visualización de un conjunto de cambios

1. Abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation>.
2. En la barra de navegación de la parte superior de la pantalla, elija la Región de AWS.
3. En la página Pilas, seleccione el nombre de la pila que contiene el conjunto de cambios que desea ver.
4. En el panel de navegación, elija Change Sets (Conjuntos de cambios) para ver una lista con los conjuntos de cambios de la pila.
5. Elija el nombre del conjunto de cambios que desee ver.

La consola de CloudFormation lo dirigirá a la página de detalles del conjunto de cambios, donde podrá ver el momento de creación del conjunto de cambios, su estado, la información usada para generar el conjunto de cambios y un resumen de estos.

En la sección Changes (Cambios), cada fila representa un recurso que CloudFormation agregará, modificará o eliminará.

- **Add (Agregar):** crea un recurso al agregar un recurso a la plantilla de la pila.
- **Modify (Modificar):** modifica un recurso cuando se cambian sus propiedades en la plantilla de la pila.
- **Remove (Eliminar):** elimina un recurso al eliminar un recurso existente de la plantilla de la pila.

 Note

Una modificación puede provocar que se interrumpa o sustituya el recurso (recreado). Para obtener más información acerca de las conductas de actualización, consulte [Comprensión de los comportamientos de actualización de los recursos de la pila](#).

Para centrarse en determinados cambios, utilice la vista de filtro. Por ejemplo, filtrar un tipo de recurso específico, como `AWS::EC2::Instance`. Para filtrar por recurso específico, especifique su ID físico o lógico, como `myWebServer` o `i-123abcd4`.

6. En la sección Cambios, seleccione Ver detalles en la columna Cambios a nivel de propiedad para consultar los cambios en los valores que se realizaron en los recursos.
7. La consola de CloudFormation lo redirige a la página de cambios a nivel de propiedad para un recurso, en donde puede ver la configuración de la plantilla del recurso antes de la ejecución de un conjunto de cambios y una predicción de la configuración luego de la ejecución del conjunto.

En la tabla de la sección de cambios a nivel de propiedad, puede ver los valores de Ruta, Tipo de cambio, Valor previo y Valor posterior para las propiedades afectadas. En la tabla, seleccione la casilla de verificación para cada cambio que quiera destacar en las vistas Previo y Posterior de la plantilla a fin de ver los cambios que se realizarán a nivel de propiedad.

- **Agregar:** las propiedades agregadas se resaltan en color verde.
- **Modificar:** las propiedades modificadas se resaltan en color azul.
- **Eliminar:** las propiedades eliminadas se resaltan en color rojo.

View a change set for nested stack (console)

Ver un conjunto de cambios para pilas anidadas (consola)

1. Abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation>.
2. En la barra de navegación de la parte superior de la pantalla, elija la Región de AWS.
3. En la página Pilas, seleccione el nombre de la pila que contiene el conjunto de cambios que desea ver.
4. En el panel de navegación, seleccione Conjuntos de cambios para ver una lista con los conjuntos de cambios de la pila.
5. Elija el nombre del conjunto de cambios que desee ver.

La consola de CloudFormation lo dirigirá a la página de detalles del conjunto de cambios, donde podrá ver el momento de creación del conjunto de cambios, su estado, la información usada para generar el conjunto de cambios y un resumen de estos.

En la sección Changes (Cambios), cada fila representa un recurso que CloudFormation agregará, modificará, eliminará o mostrará el estado dinámico.

- Add (Agregar): crea un recurso al agregar un recurso a la plantilla de la pila.
- Modify (Modificar): modifica un recurso cuando se cambian sus propiedades en la plantilla de la pila.
- Remove (Eliminar): elimina un recurso al eliminar un recurso existente de la plantilla de la pila.
- Dynamic (Dinámico): no se puede determinar la acción exacta de cambio de recursos a partir de la plantilla de la pila anidada.

Note

Una modificación puede provocar que se interrumpa o sustituya el recurso (recreado). Para obtener más información acerca de las conductas de actualización, consulte [Comprensión de los comportamientos de actualización de los recursos de la pila](#).

Para centrarse en determinados cambios, utilice la vista de filtro. Por ejemplo, filtrar un tipo de recurso específico, como **AWS::CloudFormation::Stack**. Para filtrar por recurso específico, especifique su ID físico o lógico, como **DeadLetterQueue** o **NestedStack**.

6. En la sección Changes (Cambios) , elija View nested change set (Ver conjunto de cambios anidados) del conjunto de cambios anidados que desea ver.

La consola de CloudFormation lo dirige a la página de detalles del conjunto de cambios anidado. Puede elegir Go to root change set (Ir al conjunto de cambios raíz) para ver el conjunto de cambios raíz o, puede elegir View parent change set (Ver conjunto de cambios principal) para ver el conjunto de cambios principal. Para obtener más información, consulte [Conjuntos de cambios para pilas anidadas](#).

 Note

Los conjuntos de cambios a nivel de propiedad de CloudFormation no resuelven las referencias entre pilas al crear conjuntos de cambios para pilas anidadas. Los conjuntos de cambios pueden marcar los recursos de una pila secundaria para su reemplazo condicional si hacen referencia a la salida de una pila principal y esta última se ha modificado

Ver un conjunto de cambios (AWS CLI)

1. Para obtener el ID del conjunto de cambios, ejecute el comando [change-sets](#).

Especifique el nombre de la pila que tenga el conjunto de cambios que desea ver, tal y como se muestra en el ejemplo siguiente:

```
aws cloudformation list-change-sets --stack-name MyStack
```

CloudFormation devuelve una lista de conjuntos de cambios, similar a esta:

```
{
  "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. Ejecute el comando [describe-change-set](#), especificando el ID del conjunto de cambios que desea ver. Por ejemplo:

```

aws cloudformation describe-change-set \
  --change-set-name arn:aws:cloudformation:us-east-1:123456789012:changeSet/SampleChangeSet/1a2345b6-0000-00a0-a123-00abc0abc000

```

CloudFormation devuelve información sobre el conjunto de cambios especificado.

```
{
```

```

    "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"
}

```

Utilice `--include-property-values` con `describe-change-set` para enumerar los cambios a nivel de propiedad.

La clave `Changes` enumera los cambios de los recursos. Si fuera a ejecutar este conjunto de cambios, CloudFormation actualizaría las etiquetas de la instancia de EC2 `i-1abc23d4`. Para obtener una descripción de cada campo, consulte [Change](#) el tipo de datos en la referencia de la API de AWS CloudFormation.

Para obtener más ejemplos de conjuntos de cambio, consulte [Ejemplo de conjuntos de cambio para pilas de CloudFormation](#).

Consulta de los cambios a nivel de propiedad en un conjunto de cambios (AWS CLI)

- El comando que figura a continuación muestra los cambios a nivel de propiedad relacionados con un conjunto de cambios para un recurso `AWS::EC2::NetworkInterface` que elimina la propiedad `Ipv4Prefixes`, modifica la `Description` del recurso y agrega una `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

```

A continuación, se muestra un ejemplo del resultado.

```

"ChangeSetName": "ExampleChangeSet",
  "ChangeSetId": "arn:aws:cloudformation:us-east-1:803642222207:changeSet/
ExampleChangeSet/9f7b541b-126b-44f7-998e-932174557841",
  "StackId": "arn:aws:cloudformation:us-east-1:803642222207:stack/ExampleStack/
ab664180-f686-11ee-9e29-12cd92393671",
  "StackName": "ExampleStack",
  "Description": null,
  "Parameters": null,
  "CreationTime": "2024-04-09T18:04:59.935000+00:00",
  "ExecutionStatus": "AVAILABLE",

```

```

    "Status": "CREATE_COMPLETE",
    "StatusReason": null,
    "NotificationARNs": [],
    "RollbackConfiguration": {
        "RollbackTriggers": []
    },
    "Capabilities": [],
    "Tags": null,
    "ParentChangeSetId": null,
    "IncludeNestedStacks": true,
    "RootChangeSetId": null,
    "OnStackFailure": null,
  {
    "Changes": [
      {
        "Type": "Resource",
        "ResourceChange": {
          "Action": "Modify",
          "LogicalResourceId":
"EC2NetworkInterface00eni067fd35b649a05b7100Tpyls",
          "PhysicalResourceId": "eni-067fd35b649a05b71",
          "ResourceType": "AWS::EC2::NetworkInterface",
          "Replacement": "False",
          "Scope": [
            "Properties",
            "Tags"
          ],
          "Details": [
            {
              "Target": {
                "Attribute": "Properties",
                "Name": "Ipv4Prefixes",
                "RequiresRecreation": "Never",
                "Path": "/Properties/Ipv4Prefixes",
                "BeforeValue": "[]",
                "AttributeChangeType": "Remove"
              },
              "Evaluation": "Static",
              "ChangeSource": "DirectModification"
            },
            {
              "Target": {
                "Attribute": "Properties",
                "Name": "Description",

```

```

        "RequiresRecreation": "Never",
        "Path": "/Properties/Description",
        "BeforeValue": "",
        "AfterValue": "Description",
        "AttributeChangeType": "Modify"
    },
    "Evaluation": "Static",
    "ChangeSource": "DirectModification"
},
{
    "Target": {
        "Attribute": "Tags",
        "RequiresRecreation": "Never",
        "Path": "/Properties/Tags/0",
        "AfterValue": "{\"Key\":\"Test\",\"Value\":\"Test\"}",
        "AttributeChangeType": "Add"
    },
    "Evaluation": "Static",
    "ChangeSource": "DirectModification"
}
],
    "BeforeContext": "{\"Properties\":{\"Description\":\"Description\",
    \"PrivateIpAddress\":\"172.31.76.2\", \"PrivateIpAddresses\": [{\"PrivateIpAddress
    \": \"172.31.76.2\", \"Primary\": \"true\"}], \"SecondaryPrivateIpAddressCount\": \"0\",
    \"Ipv6PrefixCount\": \"0\", \"Ipv4Prefixes\": [], \"Ipv4PrefixCount\": \"0\", \"GroupSet
    \": [\"sg-05a45689b1059e82d\"], \"Ipv6Prefixes\": [], \"SubnetId\": \"subnet-455e8969\",
    \"SourceDestCheck\": \"true\", \"InterfaceType\": \"interface\", \"Tags\": []},
    \"UpdateReplacePolicy\": \"Retain\", \"DeletionPolicy\": \"Retain\"}",
    "AfterContext": "{\"Properties\":{\"Description\":\"Description\",
    \"PrivateIpAddress\":\"172.31.76.2\", \"PrivateIpAddresses\": [{\"PrivateIpAddress
    \": \"172.31.76.2\", \"Primary\": \"true\"}], \"SecondaryPrivateIpAddressCount
    \": \"0\", \"Ipv6PrefixCount\": \"0\", \"Ipv4PrefixCount\": \"0\", \"GroupSet\":
    [\"sg-05a45689b1059e82d\"], \"Ipv6Prefixes\": [], \"SubnetId\": \"subnet-455e8969\",
    \"SourceDestCheck\": \"true\", \"InterfaceType\": \"interface\", \"Tags\": [{\"Value\":
    \"Test\", \"Key\": \"Test\"}]}, \"UpdateReplacePolicy\": \"Retain\", \"DeletionPolicy\":
    \"Retain\"}"
    }
}
],
    "ChangeSetName": "ExampleChangeSet",
    "ChangeSetId": "arn:aws:cloudformation:us-east-1:123456789012:changeSet/
    ExampleChangeSet/9f7b541b-126b-44f7-998e-932174557841",

```

```
"StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/ExampleStack/ab664180-f686-11ee-9e29-12cd92393671",
```

Uso de conjuntos de cambios con reconocimiento de desviación

Los conjuntos de cambios con reconocimiento de desviación son conjuntos de cambios de CloudFormation mejorados que le permiten identificar y gestionar la desviación de las pilas de forma segura. Las pilas se desvían de la configuración de plantilla esperada cuando los recursos de la pila se modifican fuera de CloudFormation, mediante la Consola de administración de AWS, la AWS CLI o el SDK de los servicios subyacentes. Los conjuntos de cambios con reconocimiento de desviación comparan las plantillas con el estado real de los recursos de la pila y ajustan los recursos desviados a las definiciones de sus plantillas. Cuando la definición de la plantilla de un recurso se actualiza para que coincida con su estado real, los conjuntos de cambios con reconocimiento de desviación restablecen el estado de desviación del recurso sin modificarlo.

Entre las ventajas de utilizar conjuntos de cambios con reconocimiento de desviación se incluyen las siguientes:

- Obtenga una vista previa de las sobrescrituras de las desviaciones: detecte si una implementación revertirá los cambios críticos fuera de banda realizados durante la respuesta a un incidente.
- Conciliación sistemática de las desviaciones: alinee de forma segura los recursos desviados con las definiciones de plantillas compatibles, ya sea sobrescribiendo las propiedades desviadas con valores de plantilla o actualizando la plantilla para que coincida con el estado real de los recursos.
- Capacidades de reversión mejoradas: si una implementación falla, restaure los recursos a su estado real anterior a la implementación, no al estado de la plantilla anterior.
- Visibilidad completa: compruebe exactamente cómo afectará su implementación a las configuraciones de recursos reales antes de realizar cambios.

Los conjuntos de cambios tradicionales permiten comparar la nueva plantilla con la plantilla anterior de una pila, pero no tienen en cuenta la desviación de la pila. Los conjuntos de cambios con reconocimiento de desviación resuelven este problema al proporcionar una comparación a tres bandas entre:

- Estado actual: la configuración en tiempo real de sus recursos. CloudFormation leerá la configuración de los recursos de su cuenta en el momento de la creación del conjunto de cambios mediante las API de servicio subyacentes.

- Estado de implementación anterior: la configuración definida en la plantilla de la última implementación de CloudFormation.
- Estado deseado: la configuración definida en la nueva plantilla.

Los conjuntos de cambios con reconocimiento de desviación actualizarán el estado real de todos los recursos de la pila para que coincidan con el estado deseado, incluso si un recurso no se ha modificado explícitamente en la plantilla.

Consideraciones

- Compatibilidad con los tipos de recursos: los conjuntos de cambios con reconocimiento de desviación permiten comparar el estado deseado con el estado real de cientos de tipos de recursos. En el caso de los tipos de recursos no compatibles, los conjuntos de cambios con reconocimiento de desviación recurren a la comparación del estado de implementación anterior con el estado deseado. Para obtener más información, consulte la sección Tipos de recursos admitidos.
- Propiedades de solo escritura: en el caso de las propiedades que contienen información confidencial (contraseñas, secretos), los conjuntos de cambios con reconocimiento de desviación se comparan con los valores de implementación anteriores en lugar de con los valores reales.
- Propiedades administradas por AWS: los conjuntos de cambios con reconocimiento de desviación conservan los cambios fuera de banda realizados por los servicios de AWS en las propiedades administradas, como la capacidad deseada de un grupo de escalado automático. Para obtener más información, consulte la sección de propiedades administradas por AWS.
- Teclas de etiquetas externas: los conjuntos de cambios con reconocimiento de desviación no eliminan ni modifican las claves de etiquetas que no estén especificadas en la plantilla, lo que evita conflictos con los sistemas de control de acceso basado en atributos (ABAC).
- Sustitución de recursos desviados: los conjuntos de cambios con reconocimiento de desviación no admiten la conciliación de desviaciones en el caso de propiedades inmutables.
- Archivos adjuntos entre pilas: algunos tipos de recursos, como `AWS::IAM::Policy`, pueden influir en el estado real de otros tipos de recursos, como `AWS::IAM::Role`. Los conjuntos de cambios con reconocimiento de desviación gestionan la unión de los recursos dentro de una pila. Si un recurso se modifica adjuntando un recurso secundario de una pila diferente, los conjuntos de cambios con reconocimiento de desviación detectarán la modificación como una desviación y podrán revertir el adjunto. Entre los recursos adjuntables más populares se incluyen `AWS::IAM::Policy`, `AWS::IAM::ManagedPolicy`, `AWS::EC2::SecurityGroupIngress` y `AWS::EC2::SecurityGroupEgress`.

Propiedades administradas por AWS

Puede configurar propiedades de recursos específicas para la administración activa de AWS. Por ejemplo, puede permitir que Amazon Relational Database Service (Amazon RDS) actualice automáticamente la versión secundaria del motor de una tabla de Amazon RDS. Estos cambios pueden aparecer como una desviación de la pila en CloudFormation. Los conjuntos de cambios con reconocimiento de desviación reconocen que se espera una desviación en el caso de las propiedades administradas por AWS y dejan intacto su valor real si no se ha modificado la propiedad en su plantilla. Los principales ejemplos de propiedades administradas por AWS son:

- Habilitar la propiedad `AutoMinorVersionUpgrade` de una tabla de Amazon RDS para permitir las actualizaciones automáticas de la versión del motor.
- Uso del recurso `AWS::ApplicationAutoScaling::ScalableTarget` para habilitar el escalado automático de propiedades como las unidades de capacidad de lectura/escritura de una tabla de Amazon DynamoDB y el recuento deseado de un clúster de Amazon Elastic Container Service.
- Uso de `AWS::AutoScaling::ScalingPolicy` para grupos de escalado automático.

Los conjuntos de cambios con reconocimiento de desviación aclaran las propiedades que se identificaron como administradas por AWS. Para obtener más información, consulte la sección **AWS CLI**.

Uso de conjuntos de cambios con reconocimiento de desviación (consola)

Puede crear y administrar conjuntos de cambios con reconocimiento de desviación a través de la consola de CloudFormation mediante el mismo flujo de trabajo que los conjuntos de cambios tradicionales, con opciones adicionales para los modos de implementación.

Creación de conjuntos de cambios con reconocimiento de desviación

Creación de un conjunto de cambios con reconocimiento de desviación (consola)

1. Inicie sesión en la Consola de administración de AWS y abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation>.
2. En la página Pilas, seleccione la pila en ejecución que desea actualizar.
3. Seleccione Acciones de la pila y, a continuación, elija Crear conjunto de cambios para la pila actual.

4. En la página Crear conjunto de cambios, en Tipo de conjunto de cambios, elija Conjunto de cambios con reconocimiento de desviación.
5. Elija Create change set (Crear conjunto de cambios).
6. En la página de detalles del conjunto de cambios, revise la comparación de tres modos que muestra la implementación real, la implementación anterior y el estado deseado para cada recurso afectado.
7. Si está satisfecho con los cambios, elija Ejecutar conjunto de cambios.

Revisión de conjuntos de cambios con reconocimiento de desviación

Al ver un conjunto de cambios con reconocimiento de desviación en la consola, verá información mejorada en comparación con los conjuntos de cambios tradicionales:

- Estado de desviación de la pila: indica si la pila se ha desviado desde su última implementación.
- Comparación de propiedades: muestra una diferencia en JSON entre el estado real y el deseado de un recurso afectado.
- Indicadores de desviación: marca claramente las propiedades dentro de la diferencia de JSON que se han desviado. Haga clic en Ver desviación para ver el valor de implementación anterior de una propiedad.
- Indicadores de origen de valor: muestran si los valores anteriores de una propiedad provienen del estado real o de un estado de implementación anterior.

Uso de conjuntos de cambios con reconocimiento de desviación (AWS CLI)

Puede crear y gestionar conjuntos de cambios con reconocimiento de desviación con la AWS CLI mediante la adición del parámetro `--deployment-mode REVERT_DRIFT` al comando `create-change-set`.

Creación de conjuntos de cambios con reconocimiento de desviación

Creación de un conjunto de cambios con reconocimiento de desviación

Utilice el comando [create-change-set](#) con el parámetro `--deployment-mode REVERT_DRIFT` para crear un conjunto de cambios con reconocimiento de desviación.

```
aws cloudformation create-change-set \  
  --stack-name my-stack \  
  --change-set-name my-drift-aware-changeset \  
  --deployment-mode REVERT_DRIFT
```

```
--template-body file://updated-template.yaml \  
--deployment-mode REVERT_DRIFT \  
--capabilities CAPABILITY_IAM
```

Revisión de conjuntos de cambios con reconocimiento de desviación

Revisión de los detalles de un conjunto de cambios con reconocimiento de desviación

Utilice el comando [describe-change-set](#) para revisar los detalles de un conjunto de cambios con reconocimiento de desviación:

```
aws cloudformation describe-change-set \  
--change-set-name my-drift-aware-changeset \  
--stack-name my-stack
```

La respuesta incluye información mejorada para los conjuntos de cambios con reconocimiento de desviación:

- **StackDriftStatus**: muestra si la pila se ha desviado (DRIFTED, IN_SYNC, NOT_CHECKED o UNKNOWN).
- **ResourceDriftStatus**: muestra el estado de desviación de cada recurso (DELETED, MODIFIED, IN_SYNC o NOT_CHECKED).
- **BeforeValueFrom**: indica si el valor anterior de una propiedad proviene de ACTUAL_STATE o PREVIOUS_DEPLOYMENT_STATE.
- **Drift**: contiene los detalles de desviación de una propiedad, incluidos PreviousValue, ActualValue y DriftDetectionTimestamp.
- **ResourceDriftIgnoredProperties**: contiene las propiedades de un recurso para las que el conjunto de cambios no revertirá la desviación y las razones por las que se ignora la desviación.

Para obtener más información, consulte [DescribeChangeSet](#) en la referencia de la API de AWS CloudFormation.

Ejecución de conjuntos de cambios con reconocimiento de desviación

Ejecución de un conjunto de cambios con reconocimiento de desviación

Tras revisar el conjunto de cambios, utilice el comando [execute-change-set](#) para aplicar los cambios:

```
aws cloudformation execute-change-set \  

```

```
--change-set-name my-drift-aware-changeset \  
--stack-name my-stack
```

Si se produce un error en la implementación, CloudFormation restablecerá los recursos a su estado anterior a la implementación y conservará los cambios fuera de banda realizados desde la última implementación.

Tipos de recursos admitidos

Los conjuntos de cambios con reconocimiento de desviación permiten una comparación de tres modos del estado real, el estado de la implementación anterior y el estado deseado para todos los [tipos de recursos](#) que admiten la detección de desviaciones, excepto los siguientes tipos de recursos:

- `AWS::SageMaker::MlflowTrackingServer`
- `AWS::Route53Resolver::FirewallRuleGroup`
- `AWS::MediaLive::Multiplexprogram`
- `AWS::VpcLattice::ResourceConfiguration`
- `AWS::S3::StorageLensGroup`
- `AWS::Bedrock::AgentAlias`
- `AWS::MSK::Cluster`
- `AWS::RDS::DBProxy`
- `AWS::Redshift::ClusterParameterGroup`
- `AWS::QBusiness::Index`
- `AWS::NetworkManager::CoreNetwork`
- `AWS::IAM::OIDCProvider`
- `AWS::Organizations::ResourcePolicy`
- `AWS::SNS::TopicInlinePolicy`
- `AWS::Route53::KeySigningKey`
- `AWS::DataZone::PolicyGrant`
- `AWS::Transfer::Certificate`
- `AWS::SageMaker::ImageVersion`
- `AWS::Neptune::DBParameterGroup`
- `AWS::ODB::CloudVmCluster`
- `AWS::RolesAnywhere::TrustAnchor`

- `AWS::Detective::Graph`
- `AWS::Maester::DocumentType`
- `AWS::SageMaker::ModelPackageGroup`
- `AWS::S3Express::BucketPolicy`
- `AWS::Panorama::PackageVersion`
- `AWS::S3Tables::TableBucketPolicy`

En el caso de los recursos que no admiten la comparación de tres mods, los conjuntos de cambios con reconocimiento de desviación recurren a una comparación entre el estado anterior de la implementación y el estado deseado.

Ejecución de un conjunto de cambios para una pila de CloudFormation

Para realizar los cambios descritos en un conjunto de cambios en la pila, ejecute el conjunto de cambios.

Important

Después de ejecutar el conjunto de cambios, CloudFormation elimina todos los conjuntos asociados a la pila, ya que no son válidos para la pila actualizada. Si una actualización falla, se debe crear un nuevo conjunto de cambios.

Políticas de pila y ejecución de un conjunto de cambios

Si ejecuta un conjunto de cambios en una pila que tiene asociada una política de pila, CloudFormation impone la política cuando se actualiza la pila. No se puede especificar una política de pila temporal que supere la política existente al ejecutar un conjunto de cambios. Para actualizar un recurso protegido, debe actualizar la política de pila o utilizar el método de actualización directa. Para obtener más información, consulte [Actualizar pilas directamente](#).

Execute a change set (console)

Ejecución de un conjunto de cambios

1. Abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation>.
2. En la barra de navegación de la parte superior de la pantalla, elija la Región de AWS.

3. En la página Pilas, seleccione el nombre de la pila que desea actualizar.
4. En el panel de navegación, seleccione Conjuntos de cambios para ver una lista con los conjuntos de cambios de la pila.
5. Elija el nombre del conjunto de cambios que desea ejecutar.
6. En la página de detalles del conjunto de cambios, seleccione Ejecutar conjunto de cambios.

CloudFormation inicia de manera inmediata la actualización de la pila. La consola de CloudFormation lo dirigirá a la pestaña Eventos, donde puede monitorizar el progreso de la actualización de la pila. Para obtener más información, consulte [Monitorizar el progreso de la pila](#).

Execute a change set for nested stacks (console)

Ejecución de un conjunto de cambios para las pilas anidadas

1. Abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation>.
2. En la barra de navegación de la parte superior de la pantalla, elija la Región de AWS.
3. En la página Pilas, seleccione el nombre de la pila que desea actualizar. Debe seleccionar el nombre de la pila asociado al conjunto de cambios raíz.
4. En el panel de navegación, seleccione Conjuntos de cambios para ver una lista con los conjuntos de cambios de la pila.
5. Seleccione el nombre del conjunto de cambios raíz que desea ejecutar.
6. En la página de detalles del conjunto de cambios, seleccione Ejecutar conjunto de cambios.

Note

Si se seleccionó Habilitado para los conjuntos de cambios de las pilas anidadas durante el proceso de [Creación de un conjunto de cambios para una pila de CloudFormation](#), CloudFormation ejecutará los cambios descritos en el conjunto de cambios raíz y los conjuntos de cambios anidados.

CloudFormation inicia de manera inmediata la actualización de la pila. La consola de CloudFormation lo dirigirá a la pestaña Eventos, donde puede monitorizar el progreso de la actualización de la pila. Para obtener más información, consulte [Monitorizar el progreso de la pila](#).

Ejecución de un conjunto de cambios (AWS CLI)

- Ejecute el comando [execute-change-set](#).

Especifique el ID del conjunto de cambios que desea ejecutar, tal y como se muestra en el ejemplo siguiente:

```
aws cloudformation execute-change-set \  
  --change-set-name \  
    arn:aws:cloudformation:us-east-1:123456789012:changeSet/  
SampleChangeSet/1a2345b6-0000-00a0-a123-00abc0abc000
```

El comando del ejemplo ejecuta un conjunto de cambios con el ID `arn:aws:cloudformation:us-east-1:123456789012:changeSet/SampleChangeSet/1a2345b6-0000-00a0-a123-00abc0abc000`.

Después de ejecutar el comando, CloudFormation inicia la actualización de la pila. Para ver el progreso de la pila, utilice el comando [describe-stacks](#).

Eliminación de un conjunto de cambios para una pila de CloudFormation

Cuando se elimina un conjunto de cambios, también se elimina de la lista de conjuntos de cambios de la pila. La eliminación de un conjunto de cambios impide que usted u otro usuario ejecuten por accidente un conjunto de cambios que no debería aplicarse. CloudFormation conserva todos los conjuntos de cambios hasta que actualice la pila, a menos que los elimine.

Delete a change set

Eliminación de un conjunto de cambios (consola)

1. Abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation>.
2. En la barra de navegación de la parte superior de la pantalla, elija la Región de AWS.
3. En la página Pilas, seleccione el nombre de la pila que contenga el conjunto de cambios que desea eliminar.
4. En el panel de navegación, seleccione Conjuntos de cambios para ver una lista con los conjuntos de cambios de la pila.
5. Seleccione el nombre del conjunto de cambios que desea eliminar.
6. En la página de detalles del conjunto de cambios, seleccione Eliminar conjunto de cambios.

CloudFormation comienza a eliminar de manera inmediata el conjunto de cambios de la lista de conjuntos de cambios de la pila y lo redirige a la página Stacks (Pilas).

Delete a change set for nested stacks (console)

Eliminación de un conjunto de cambios para pilas anidadas

1. Abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation>.
2. En la barra de navegación de la parte superior de la pantalla, elija la Región de AWS.
3. En la página Pilas, seleccione el nombre de pila asociado al conjunto de cambios raíz.
4. En el panel de navegación, seleccione Conjuntos de cambios para ver una lista con los conjuntos de cambios de la pila.
5. Seleccione el nombre del conjunto de cambios que desea eliminar.
6. En la página de detalles del conjunto de cambios, seleccione Delete (Eliminar). Al seleccionar Eliminar conjunto de cambios, se eliminará toda la jerarquía de conjuntos de cambios anidados.

Note

La operación de eliminación de conjuntos de cambios para pilas anidadas es asíncrona y mostrará el estado DELETE_PENDING seguido del estado DELETE_IN_PROGRESS. Una vez finalizada la operación de eliminación del conjunto de cambios, los conjuntos se eliminarán de la lista. Las pilas anidadas con el estado REVIEW_IN_PROGRESS también se eliminarán si se crearon durante la creación del conjunto de cambios.

CloudFormation comienza a eliminar de manera inmediata el conjunto de cambios de la lista de conjuntos de cambios de la pila.

Note

Si cuenta con pilas anidadas atascadas en una operación en curso, consulte Troubleshooting Errors (Solución de problemas) en [Fallo de reversión de pilas anidadas](#).

Eliminación de un conjunto de cambios (AWS CLI)

- Ejecute el comando [delete-change-set](#), en el que especifique el ID del conjunto de cambios que desea eliminar, tal y como se muestra en el ejemplo siguiente:

```
aws cloudformation delete-change-set \  
  --change-set-name \  
    arn:aws:cloudformation:us-east-1:123456789012:changeSet/  
SampleChangeSet/1a2345b6-0000-00a0-a123-00abc0abc000
```

Ejemplo de conjuntos de cambio para pilas de CloudFormation

En esta sección, se brindan ejemplos de los conjuntos de cambios que CloudFormation crearía para cambios de pila comunes. Muestran cómo editar una plantilla de manera directa, modificar un único parámetro de entrada, planificar la recreación de recursos (sustituciones) para evitar la pérdida de los datos sin copia de seguridad ni interrumpir las aplicaciones que se ejecuten en la pila y añadir y eliminar recursos. Para ilustrar cómo funcionan los conjuntos de cambios, le guiaremos por los cambios enviados y debatiremos el conjunto de cambios resultante. Dado que cada ejemplo se basa en el ejemplo anterior y se asume que usted lo entiende, le recomendamos que los lea en orden. Para obtener una descripción de cada campo de un conjunto de cambios, consulte el tipo de datos [Change \(Cambio\)](#) en la referencia de la API de AWS CloudFormation.

Puede utilizar la [consola](#), la AWS CLI, o la operación [DescribeChangeSet](#) de API de CloudFormation para ver los detalles del conjunto de cambios.

Generamos cada uno de los conjuntos de cambios desde una pila con la siguiente [plantilla de ejemplo](#):

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

```

Modificación directa de una plantilla

Cuando se modifican directamente los recursos de la plantilla de pila para generar un conjunto de cambios, CloudFormation clasifica el cambio como una modificación directa, en lugar de cambios iniciados por un valor de parámetros actualizados. El siguiente conjunto de cambios, que añadió una nueva etiqueta a la instancia `i-1abc23d4`, es un ejemplo de modificación directa. Todos los demás valores de entrada, como los valores de los parámetros y capacidades, no cambian, por lo que vamos a centrarnos en la estructura de `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"  
}
```

En la estructura de `Changes`, solo hay una estructura de `ResourceChange`. Esta estructura describe información como el tipo de recurso que cambiará CloudFormation, la acción que realizará CloudFormation, el ID del recurso, el alcance del cambio y si el cambio requiere una sustitución (donde CloudFormation crea un nuevo recurso y, a continuación, elimina el anterior). En el ejemplo, el conjunto de cambios indica que CloudFormation va a modificar el atributo `Tags` de la instancia de EC2 `i-1abc23d4` y no requiere que se sustituya la instancia.

En la estructura de `Details`, CloudFormation etiqueta este cambio como una modificación directa que nunca necesitará la recreación (sustitución) de la instancia. Puede ejecutar con confianza este cambio con la confianza de que CloudFormation no sustituirá la instancia.

CloudFormation muestra este cambio como una evaluación `Static`. Una evaluación estática significa que CloudFormation puede determinar el valor de la etiqueta antes de ejecutar el conjunto de cambios. En algunos casos, CloudFormation puede determinar un valor solo después de ejecutar un conjunto de cambios. CloudFormation etiqueta esos cambios como evaluaciones de `Dynamic`. Por ejemplo, si hace referencia a un recurso actualizado que se sustituye condicionalmente, CloudFormation no puede determinar si cambiará la referencia al recurso actualizado.

Modificación de un valor de parámetro de entrada

Si modifica un valor de parámetro de entrada, CloudFormation genera dos cambios para cada recurso que utilice el valor del parámetro actualizado. En este ejemplo, queremos destacar cómo son dichos cambios y en qué información debería centrarse. El siguiente ejemplo se generó al cambiar solo el valor del parámetro de entrada `Purpose`.

El parámetro `Purpose` especifica un valor de clave de etiqueta para la instancia de EC2. En el ejemplo, se cambió el valor del parámetro de `testing` a `production`. El nuevo valor aparece en la estructura de `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"  
}
```

La estructura de Changes funciona de forma similar al ejemplo de [Modificación directa de una plantilla](#). Sólo hay una estructura de ResourceChange; describe un cambio en el atributo Tags de la instancia de EC2 i-1abc23d4.

Sin embargo, en la estructura Details, el conjunto de cambios muestra dos cambios para el atributo Tags, aunque solo se cambió el valor de un parámetro. Los recursos que hacen referencia a un valor de parámetro cambiado (con la función intrínseca Ref) siempre generan dos cambios: uno con una Dynamic y otro con una evaluación Static. Estos tipos de cambios son visibles al consultar los siguientes campos:

- Para el cambio de evaluación Static, consulte el campo ChangeSource. En este ejemplo, el campo ChangeSource equivale a ParameterReference, lo que significa que este cambio es un resultado de un valor de referencia de parámetro actualizado. El conjunto de cambios debe contener un cambio de evaluación Dynamic similar.
- Puede encontrar el cambio de evaluación Dynamic coincidente al comparar la estructura de Target en ambos cambios, que contendrá la misma información. En este ejemplo, las estructuras Target de ambos cambios contienen los mismos valores para los campos Attribute y RequireRecreation.

Para estos tipos de cambios, céntrese en la evaluación estática, la cual ofrece más información detallada sobre el cambio. En este ejemplo, la evaluación estática muestra que el cambio es el resultado de un cambio en un valor de referencia de parámetro (ParameterReference). El campo CauseEntity (el parámetro Purpose) indica el parámetro exacto que se cambió.

Determinación del valor del campo de repuesto

El campo Replacement de una estructura de ResourceChange indica si CloudFormation volverá a crear el recurso. La planificación de la recreación de los recursos (sustituciones) impide la pérdida de datos sin copia de seguridad o la interrupción de aplicaciones que se ejecutan en la pila.

El valor del campo `Replacement` depende de si un cambio requiere una sustitución, lo cual se indica en el campo `RequiresRecreation` en la estructura `Target` de un cambio. Por ejemplo, si el campo `RequiresRecreation` es `Never`, el campo `Replacement` es `False`. Sin embargo, si hay varios cambios en un solo recurso y cada cambio tiene un valor diferente para el campo `RequiresRecreation`, CloudFormation actualiza el recurso con el comportamiento más intrusivo. En otras palabras, si solo uno de los tantos cambios requiere una sustitución, CloudFormation debe sustituir el recurso y, por lo tanto, establece el campo `Replacement` en `True`.

Se generó el siguiente conjunto de cambios al cambiar los valores de cada parámetro (`Purpose`, `InstanceType` y `KeyPairName`), que utiliza la instancia de EC2. Con estos cambios, será necesario que CloudFormation sustituya la instancia, ya que el campo `Replacement` es igual a `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"
}

```

Visualice cada cambio (las evaluaciones estáticas de la estructura de `Details`) para identificar el cambio que requiere sustituir el recurso. En este ejemplo, cada cambio tiene un valor diferente para el campo `RequireRecreation`, pero el cambio de la propiedad `KeyName` tiene el comportamiento de actualización más intrusivo, siempre requiere una recreación. CloudFormation sustituirá la instancia, ya que se cambió el nombre de clave.

Si no se cambió el nombre de clave, el cambio de la propiedad `InstanceType` tendría el comportamiento de actualización más intrusivo (`Conditionally`), por lo que el campo `Replacement` sería `Conditionally`. Para encontrar las condiciones en las que CloudFormation sustituye la instancia, consulte el comportamiento de la actualización de la propiedad `InstanceType` del tipo de recurso [AWS::EC2::Instance](#).

Adición y eliminación de recursos

El siguiente ejemplo se ha generado al enviar una plantilla modificada que elimina la instancia de EC2 y añade un grupo de Auto Scaling y una configuración de lanzamiento.

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

```

        "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"
}

```

En la estructura de Changes, existen tres estructuras de ResourceChange, una para cada recurso. Para cada recurso, el campo Action indica si CloudFormation agrega o elimina el recurso. Los campos Scope y Details están vacíos, ya que se aplican únicamente a los recursos modificados.

Para nuevos recursos, CloudFormation no puede determinar el valor de algunos campos hasta que se ejecute el cambio. Por ejemplo, CloudFormation no proporciona el ID físico del grupo de escalado automático y la configuración de lanzamiento, ya que no existen todavía. CloudFormation crea los nuevos recursos cuando se ejecuta el conjunto de cambios.

Consulta de cambios a nivel de propiedad

En el ejemplo que figura a continuación, se muestran los cambios a nivel de propiedad de la propiedad Tag de una instancia de Amazon EC2. La etiqueta Value y Key cambiarán a Test.

```

"ChangeSetName": "SampleChangeSet",
"ChangeSetId": "arn:aws:cloudformation:us-east-1:123456789012:changeSet/
SampleChangeSet/38d91d27-798d-4736-9bf1-fb7c46207807",
"StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
SampleEc2Template/68edcdc0-f6b6-11ee-966c-126d572cdd11",
"StackName": "SampleEc2Template",

```

```

    "Description": "A sample EC2 instance template for testing change sets.",
    "Parameters": [
      {
        "ParameterKey": "KeyPairName",
        "ParameterValue": "BatchTest"
      },
      {
        "ParameterKey": "Purpose",
        "ParameterValue": "testing"
      },
      {
        "ParameterKey": "InstanceType",
        "ParameterValue": "t2.micro"
      }
    ],
    "CreationTime": "2024-04-09T21:29:10.759000+00:00",
    "ExecutionStatus": "AVAILABLE",
    "Status": "CREATE_COMPLETE",
    "StatusReason": null,
    "NotificationARNs": [],
    "RollbackConfiguration": {
...skipping...
    {
      "Changes": [
        {
          "Type": "Resource",
          "ResourceChange": {
            "Action": "Modify",
            "LogicalResourceId": "MyEC2Instance",
            "PhysicalResourceId": "i-0cc7856a36315e62b",
            "ResourceType": "AWS::EC2::Instance",
            "Replacement": "False",
            "Scope": [
              "Tags"
            ],
            "Details": [
              {
                "Target": {
                  "Attribute": "Tags",
                  "RequiresRecreation": "Never",
                  "Path": "/Properties/Tags/0/Value",
                  "BeforeValue": "testing",
                  "AfterValue": "Test",
                  "AttributeChangeType": "Modify"
                }
              }
            ]
          }
        }
      ]
    }
  ]
}

```

```

        },
        "Evaluation": "Static",
        "ChangeSource": "DirectModification"
    },
    {
        "Target": {
            "Attribute": "Tags",
            "RequiresRecreation": "Never",
            "Path": "/Properties/Tags/0/Key",
            "BeforeValue": "Purpose",
            "AfterValue": "Test",
            "AttributeChangeType": "Modify"
        },
        "Evaluation": "Static",
        "ChangeSource": "DirectModification"
    }
],
    "BeforeContext": "{\"Properties\":{\"KeyName\":\"BatchTest\",\"ImageId\":"
    "\"ami-8fcee4e5\",\"InstanceType\":\"t2.micro\",\"Tags\":[{\"Value\":\"testing\",
    \"Key\":\"Purpose\"}]}}",
    "AfterContext": "{\"Properties\":{\"KeyName\":\"BatchTest\",\"ImageId\":"
    "\"ami-8fcee4e5\",\"InstanceType\":\"t2.micro\",\"Tags\":[{\"Value\":\"Test\", \"Key\":"
    "\"Test\"}]}}"}
    }
}
]

```

La estructura `Details` muestra los valores correspondientes de `Key` y `Value` previos a la ejecución del conjunto de cambios y su estado luego de la ejecución del conjunto.

Conjuntos de cambios para pilas anidadas

Con los conjuntos de cambios para pilas anidadas, puede previsualizar los cambios en los recursos de la aplicación y la infraestructura en toda la jerarquía de pilas anidadas y continuar con las actualizaciones cuando haya confirmado que todos los cambios son los previstos.

Consulte las siguientes secciones para obtener más detalles sobre los conjuntos de cambios para pilas anidadas:

Temas

- [Información general de conjuntos de cambios y pilas anidadas](#)
- [Trabajar con conjuntos de cambios para pilas anidadas \(consola\)](#)

- [Trabajar con conjuntos de cambios para pilas anidadas \(AWS CLI\)](#)

Información general de conjuntos de cambios y pilas anidadas

Los conjuntos de cambios para pilas anidadas combinan las siguientes características para ampliar el alcance de la vista previa de los cambios a toda la jerarquía de pila:

- Un conjunto de cambios es una capacidad de CloudFormation que ofrece una vista previa de cómo los cambios propuestos en una pila afectarán a los recursos existentes o recién creados. Al crear un conjunto de cambios, CloudFormation proporciona una lista de cambios propuestos comparando la pila con los cambios realizados en los recursos enviados. Para obtener más información sobre los conjuntos de cambios, consulte [Actualización de pilas de CloudFormation con conjuntos de cambios](#).
- Una pila anidada es una pila creada como parte de otra pila. Por ejemplo, podría tener recursos relacionados con la red y la seguridad en una pila anidada y recursos de aplicaciones en otra. Particionar los modelos de aplicaciones de esta manera ayuda con el mantenimiento y la reutilización del código. Para obtener más información sobre las pilas anidadas, consulte [División de una plantilla en piezas reutilizables utilizando pilas anidadas](#).

Trabajar con conjuntos de cambios para pilas anidadas (consola)

- Crear un conjunto de cambios – Cree un juego de cambios enviando cambios desde cualquier nivel de la jerarquía de pila. Puede enviar una plantilla de pila modificada o valores de parámetros de entrada modificados y CloudFormation comparará la pila anidada con los cambios enviados para generar un conjunto de cambios. Los conjuntos de cambios para pilas anidadas están habilitados de forma predeterminada en la consola CloudFormation. Para obtener más información, consulte [Creación de un conjunto de cambios para una pila de CloudFormation](#).

▼ 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

Un conjunto de cambios raíz es el conjunto de cambios asociado a la pila a partir de la que se crea toda la jerarquía de conjuntos de cambios. Debe ejecutar o eliminar conjuntos de cambios para pilas anidadas desde el conjunto de cambios raíz. Para obtener más información, consulte [Operaciones de apilamiento en pilas anidadas](#).

- Ver el conjunto de cambios – Visualice los cambios en los recursos dentro de pilas anidadas antes de ejecutarlos. Puede ver los cambios propuestos en la sección Cambios del conjunto de cambios navegando por la pila actual y sus conjuntos de cambios anidados. Para obtener más información, consulte [Visualización de un conjunto de cambios para una pila de CloudFormation](#).
- Ejecutar el conjunto de cambios – Ejecute los cambios descritos en el conjunto de cambios que pertenecen a la pila actual y sus descendientes. La operación de ejecución debe llevarse a cabo desde el conjunto de cambios raíz. Para obtener más información, consulte [Ejecución de un conjunto de cambios para una pila de CloudFormation](#).
- Eliminar el conjunto de cambios – Elimina los conjuntos de cambios de la pila actual. La eliminación de un conjunto de cambios ayuda a evitar que usted u otro usuario inicien accidentalmente un conjunto de cambios que no debe aplicarse. La operación de eliminación debe hacerse desde el conjunto de cambios raíz. Para obtener más información, consulte [Eliminación de un conjunto de cambios para una pila de CloudFormation](#).

Trabajar con conjuntos de cambios para pilas anidadas (AWS CLI)

create-change-set

- [create-change-set](#) – los conjuntos de cambios para las pilas anidadas no están habilitados de forma predeterminada para AWS CLI. Para crear un conjunto de cambios para toda la jerarquía de pila, especifique la opción `--include-nested-stacks`. Para obtener más información, consulte [Creación de un conjunto de cambios para una pila de CloudFormation](#).

El siguiente ejemplo de AWS CLI crea un conjunto de cambios para la pila raíz especificada.

```
aws cloudformation create-change-set \  
  --stack-name my-root-stack \  
  --change-set-name my-root-stack-change-set \  
  --template-body file://template.yaml \  
  --include-nested-stacks
```

```
--capabilities CAPABILITY_IAM \  
--include-nested-stacks
```

A continuación, se muestra un ejemplo del resultado.

```
{  
  "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](#) – Devuelve una lista de cambios que CloudFormation realizará si ejecuta el conjunto de cambios. Si el conjunto de cambios especificado contiene conjuntos de cambios secundarios que pertenecen a pilas anidadas, ChangeSetId devolverá información sobre ese conjunto de cambios. Para obtener más información, consulte [Visualización de un conjunto de cambios para una pila de CloudFormation](#).

El siguiente ejemplo de AWS CLI describe un conjunto de cambios para la pila raíz especificada.

```
aws cloudformation describe-change-set \  
  --change-set-name my-root-stack-change-set \  
  --stack-name my-root-stack
```

A continuación, se muestra un ejemplo del resultado.

```
{  
  "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

```

```
}
```

El siguiente ejemplo de AWS CLI describe un conjunto de cambios para la pila anidada especificada.

```
aws cloudformation describe-change-set \
  --change-set-name my-nested-stack-change-set \
  --stack-name my-nested-stack
```

A continuación, se muestra un ejemplo del resultado.

```
{
  "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](#) – Crea o actualiza una pila con la información de entrada que se proporcionó cuando se creó el conjunto de cambios especificado. Para crear un conjunto de cambios para toda la jerarquía de pila, debe especificar la opción `--include-nested-stacks` durante la operación `create-change-set`. Para obtener más información, consulte [Ejecución de un conjunto de cambios para una pila de CloudFormation](#).

Note

`execute-change-set` debe ejecutarse desde el conjunto de cambios raíz y aplicará el conjunto de cambios a toda la jerarquía de pilas.

El siguiente ejemplo de AWS CLI inicia un conjunto de cambios para la pila raíz especificada.

```
aws cloudformation execute-change-set \
```

```
--stack-name my-root-stack \  
--change-set-name my-root-stack-change-set
```

delete-change-set

- [delete-change-set](#) – Elimina el conjunto de cambios especificado. La eliminación de conjuntos de cambios garantiza que nadie ejecutará el conjunto de cambios equivocado. La eliminación de conjuntos de cambios es asincrónica para los conjuntos de cambios creados con la opción `--include-nested-stacks`. Para obtener más información, consulte [Eliminación de un conjunto de cambios para una pila de CloudFormation](#).

Note

`delete-change-set` debe ejecutarse desde el conjunto de cambios raíz y eliminará toda la jerarquía de conjuntos de cambios. Las pilas anidadas cuyo estado es `REVIEW_IN_PROGRESS` también se eliminarán si se crearon durante la operación de `create-change-set`.

El siguiente ejemplo de AWS CLI elimina un conjunto de cambios para la pila raíz especificada.

```
aws cloudformation delete-change-set \  
--stack-name my-root-stack \  
--change-set-name my-root-stack-change-set
```

Validación de las implementaciones de pilas

Con la validación previa a la implementación, puede identificar y resolver posibles problemas de implementación antes de ejecutar sus conjuntos de cambios de CloudFormation. Esta característica valida sus plantillas frente a los escenarios de error más comunes, lo que le ayuda a detectar los problemas al principio del ciclo de desarrollo.

Temas

- [Cómo funciona la validación previa a la implementación](#)
- [Consideraciones](#)
- [Requisitos previos](#)
- [Validación de la implementación de una pila \(consola\)](#)

- [Validación de la implementación de una pila \(AWS CLI\)](#)
- [Tipos de validación](#)
- [Limitaciones de recursos](#)

Cómo funciona la validación previa a la implementación

La validación previa a la implementación implica estas fases:

1. Cree su conjunto de cambios: genere un conjunto de cambios como lo haría normalmente para las actualizaciones de la pila de CloudFormation. La validación previa a la implementación está habilitada de forma predeterminada al crear el conjunto de cambios.
2. Ejecución de la validación: CloudFormation ejecuta varias comprobaciones de validación en la plantilla y el entorno de destino. Actualmente, se admiten tres tipos de validación: la validación de la sintaxis de propiedades en función de los esquemas de recursos, la detección de conflictos en los nombres de los recursos con los recursos existentes y la validación del vacío de los buckets de S3 para las operaciones de eliminación.
3. Revise los resultados de la validación: CloudFormation proporciona comentarios detallados sobre cualquier problema encontrado, incluida una ruta precisa que indica la ubicación del problema en la plantilla, lo que elimina la depuración manual de la plantilla.
4. Resuelva los problemas: para solucionar los problemas identificados, actualice las plantillas o resuelva los conflictos antes de proceder con la implementación.
5. Ejecute con confianza: implemente su conjunto de cambios sabiendo que los escenarios de error más comunes se han validado por adelantado.

Consideraciones

Cuando utilice la validación previa a la implementación, tenga en cuenta lo siguiente:

- La validación previa a la implementación se centra en los tres escenarios de error de implementación más comunes. No garantiza que la implementación se realice correctamente, pero reduce la probabilidad de que se produzcan errores comunes.
- Los modos de validación se comportan de forma diferente:
 - El modo de ERROR impide la ejecución del conjunto de cambios cuando la validación detecta errores, lo que garantiza que las plantillas problemáticas no puedan continuar con la

implementación. Esto se aplica a los errores de sintaxis de las propiedades y a los conflictos de nomenclatura de los recursos.

- El modo de ADVERTENCIA permite que la creación del conjunto de cambios se realice correctamente a pesar de los errores de validación y proporciona advertencias que los desarrolladores pueden revisar y corregir antes de ejecutarlas. Esto se aplica a las infracciones de restricciones, como el vacío de los buckets de S3, que puede resolverse mediante una intervención manual.
- Los resultados de la validación están vinculados al conjunto de cambios específico. Si modifica la plantilla, tendrá que crear un nuevo conjunto de cambios para obtener los resultados de la validación actualizados.
- La validación de los buckets de S3 solo comprueba la presencia de objetos, no las políticas de buckets u otras restricciones que puedan impedir su eliminación.

Requisitos previos

Para utilizar la validación previa a la implementación, debe tener:

- Los permisos de IAM necesarios para crear conjuntos de cambios y leer los recursos en su cuenta. Para comprobar el vacío de un bucket de S3, necesita el permiso `s3:ListBucket`.
- Acceda a las Regiones de AWS donde están implementadas sus pilas.
- Plantillas de CloudFormation que desee validar antes de la implementación.

Validación de la implementación de una pila (consola)

Utilice el siguiente procedimiento para validar la implementación de la pila mediante la consola.

Validación de una plantilla antes de la implementación

1. Inicie sesión en la Consola de administración de AWS y abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation>.
2. En la barra de navegación de la parte superior de la pantalla, elija la Región de AWS en donde se encuentra la pila.
3. En la página Pilas, seleccione la pila en ejecución para la que desea crear un conjunto de cambios.

4. En el panel de detalles de la pila, seleccione Actualizar pila y, a continuación, seleccione Crear un conjunto de cambios.
5. En la página Crear conjunto de cambios para **stack-name**, cargue la plantilla actualizada o especifique el origen de la plantilla.
6. Seleccione Siguiente para continuar con el resto de los pasos de configuración del conjunto de cambios.
7. Si la plantilla incluye recursos de IAM, en Capacidades seleccione Acepto que CloudFormation pueda crear recursos de IAM. Los recursos de IAM pueden modificar los permisos de su cuenta de AWS; revise estos recursos para garantizar que solo se le permitan las acciones que planea. Para obtener más información, consulte la sección sobre [reconocimiento de recursos de IAM en plantillas de CloudFormation](#).
8. En la página Revisar, seleccione Crear conjunto de cambios.
9. CloudFormation creará el conjunto de cambios y ejecutará las comprobaciones de validación. Revise los resultados de la validación en la pestaña Validación de la implementación.
10. Si la validación se aprueba o está satisfecho con las advertencias, seleccione Ejecutar conjunto de cambios para implementar los cambios.
11. Si se produce un error en la validación, corrija los problemas y cree un nuevo conjunto de cambios para volver a validar la implementación.

Validación de la implementación de una pila (AWS CLI)

Los comandos de la AWS CLI para la validación previa a la implementación incluyen:

- [create-change-set](#) se valida automáticamente durante la creación del conjunto de cambios.
- [describe-change-set](#) para verificar el estado del conjunto de cambios.
- [describe-events](#) para revisar los resultados de la validación.

Utilice el siguiente procedimiento para validar la implementación de la pila mediante la AWS CLI.

Validación de una plantilla antes de la implementación

1. Utilice el comando [create-change-set](#):

```
aws cloudformation create-change-set \  
  --stack-name MyStack \  
  --template-url ...
```

```
--change-set-name MyChangeSet \  
--change-set-type "CREATE" \  
--template-body file://updated-template.yaml
```

El comando devuelve tanto el ARN del conjunto de cambios como el ARN de la pila.

2. Utilice el comando [describe-events](#) con el ARN del conjunto de cambios o el nombre del conjunto de cambios para revisar el estado y los resultados de la validación.

```
aws cloudformation describe-events \  
  --change-set-name "arn:aws:cloudformation:us-east-1:123456789012:changeSet/MyChangeSet/94498df5-1afb-43b1-9869-9f82b2d877ac"
```

Ejemplo de salida de errores de validación:

```
{  
  "OperationEvents": [  
    {  
      "EventId": "9b5c9a29-4704-4ad0-8082-afb49418d55b",  
      "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/MyStack/  
c3908380-b357-11f0-a97f-0ad08f35df65",  
      "OperationId": "f558b823-e1e3-4de3-a222-e6b930ddcad4",  
      "OperationType": "CREATE_CHANGESET",  
      "OperationStatus": "FAILED",  
      "EventType": "STACK_EVENT",  
      "Timestamp": "2025-10-27T17:10:02.923Z",  
      "StartTime": "2025-10-27T17:09:57.537Z",  
      "EndTime": "2025-10-27T17:10:02.923Z"  
    },  
    {  
      "EventId": "2d8c3262-3468-4283-82fb-6e780e9e4f1d",  
      "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/MyStack/  
c3908380-b357-11f0-a97f-0ad08f35df65",  
      "OperationId": "f558b823-e1e3-4de3-a222-e6b930ddcad4",  
      "OperationType": "CREATE_CHANGESET",  
      "EventType": "VALIDATION_ERROR",  
      "LogicalResourceId": "NotificationBucket",  
      "PhysicalResourceId": "",  
      "ResourceType": "AWS::S3::Bucket",  
      "Timestamp": "2025-10-27T17:10:02.461Z",  
      "ValidationFailureMode": "FAIL",  
      "ValidationName": "PROPERTY_VALIDATION",  
      "ValidationStatus": "FAILED",  
    }  
  ]  
}
```



```
"ValidationStatusReason": "#/NotificationConfiguration/
QueueConfigurations/0: required key [Event] not found",
  "ValidationPath": "/Resources/NotificationBucket/Properties/
NotificationConfiguration/QueueConfigurations/0"
}
]
}
```

3. Para corregir los errores de validación, actualice la plantilla y, a continuación, cree un conjunto de cambios nuevo.
4. Una vez que la validación haya pasado, ejecute el conjunto de cambios:

```
aws cloudformation execute-change-set \
  --change-set-name MyChangeSet \
  --stack-name MyStack
```

Tipos de validación

La validación previa a la implementación incluye los siguientes tipos de comprobaciones:

- Validación de la sintaxis de propiedades: valida las propiedades de los recursos comparándolas con los esquemas de recursos de AWS. Comprueba las propiedades requeridas y los valores de propiedad válidos e identifica las combinaciones de propiedades obsoletas o no compatibles.
- Detección de conflictos de nombres de recursos: comprueba si hay conflictos de nomenclatura con los recursos de AWS existentes. Valida que los nombres de los recursos cumplen los requisitos de nomenclatura de AWS e identifica los posibles conflictos antes de los intentos de implementación.
- Validación del vacío de los buckets de S3: advierte cuando se intenta eliminar los buckets de S3 que contienen objetos. Proporciona recuentos de objetos para ayudar a evaluar el impacto de la eliminación y ayuda a prevenir los errores comunes de eliminación de S3.

Cada tipo de validación proporciona mensajes de error específicos y una ubicación del error en la plantilla para ayudarle a resolver los problemas rápidamente.

Limitaciones de recursos

Los siguientes tipos de recursos no se admiten para la validación previa a la implementación:

- `AWS::ApiGatewayV2::ApiGatewayManagedOverrides`

- `AWS::ApiGatewayV2::Stage`
- `AWS::AppMesh::GatewayRoute`
- `AWS::AppMesh::Mesh`
- `AWS::AppMesh::Route`
- `AWS::AppMesh::VirtualGateway`
- `AWS::AppMesh::VirtualNode`
- `AWS::AppMesh::VirtualRouter`
- `AWS::AppMesh::VirtualService`
- `AWS::AppStream::Fleet`
- `AWS::AppStream::Stack`
- `AWS::AppStream::StackFleetAssociation`
- `AWS::AppStream::StackUserAssociation`
- `AWS::AppStream::User`
- `AWS::AppSync::ApiCache`
- `AWS::AppSync::ApiKey`
- `AWS::AppSync::GraphQLSchema`
- `AWS::AutoScalingPlans::ScalingPlan`
- `AWS::Budgets::Budget`
- `AWS::CertificateManager::Certificate`
- `AWS::Cloud9::EnvironmentEC2`
- `AWS::CloudFormation::CustomResource`
- `AWS::CloudFormation::Macro`
- `AWS::CloudFormation::WaitCondition`
- `AWS::CloudFormation::WaitConditionHandle`
- `AWS::CloudFront::StreamingDistribution`
- `AWS::CloudWatch::AnomalyDetector`
- `AWS::CloudWatch::InsightRule`
- `AWS::CodeBuild::Project`

- `AWS::CodeBuild::ReportGroup`
- `AWS::CodeBuild::SourceCredential`
- `AWS::CodeCommit::Repository`
- `AWS::CodeDeploy::DeploymentGroup`
- `AWS::CodeStar::GitHubRepository`
- `AWS::Config::ConfigurationRecorder`
- `AWS::Config::DeliveryChannel`
- `AWS::Config::OrganizationConfigRule`
- `AWS::Config::RemediationConfiguration`
- `AWS::DAX::Cluster`
- `AWS::DAX::ParameterGroup`
- `AWS::DAX::SubnetGroup`
- `AWS::DirectoryService::MicrosoftAD`
- `AWS::DLM::LifecyclePolicy`
- `AWS::DMS::Certificate`
- `AWS::DMS::Endpoint`
- `AWS::DMS::EventSubscription`
- `AWS::DMS::ReplicationInstance`
- `AWS::DMS::ReplicationSubnetGroup`
- `AWS::DMS::ReplicationTask`
- `AWS::DocDB::DBCluster`
- `AWS::DocDB::DBClusterParameterGroup`
- `AWS::DocDB::DBInstance`
- `AWS::DocDB::DBSubnetGroup`
- `AWS::DocDB::EventSubscription`
- `AWS::EC2::ClientVpnAuthorizationRule`
- `AWS::EC2::ClientVpnEndpoint`
- `AWS::EC2::ClientVpnRoute`

- `AWS::EC2::ClientVpnTargetNetworkAssociation`
- `AWS::EC2::NetworkInterfacePermission`
- `AWS::ElastiCache::CacheCluster`
- `AWS::ElastiCache::ReplicationGroup`
- `AWS::ElastiCache::SecurityGroup`
- `AWS::ElastiCache::SecurityGroupIngress`
- `AWS::ElasticLoadBalancing::LoadBalancer`
- `AWS::ElasticLoadBalancingV2::ListenerCertificate`
- `AWS::Elasticsearch::Domain`
- `AWS::EMR::Cluster`
- `AWS::EMR::InstanceFleetConfig`
- `AWS::EMR::InstanceGroupConfig`
- `AWS::FSx::FileSystem`
- `AWS::FSx::Snapshot`
- `AWS::FSx::StorageVirtualMachine`
- `AWS::FSx::Volume`
- `AWS::Glue::Classifier`
- `AWS::Glue::Connection`
- `AWS::Glue::CustomEntityType`
- `AWS::Glue::DataCatalogEncryptionSettings`
- `AWS::Glue::DataQualityRuleset`
- `AWS::Glue::DevEndpoint`
- `AWS::Glue::MLTransform`
- `AWS::Glue::Partition`
- `AWS::Glue::SecurityConfiguration`
- `AWS::Glue::Table`
- `AWS::Glue::TableOptimizer`
- `AWS::Glue::Workflow`
- `AWS::Greengrass::ConnectorDefinition`

- `AWS::Greengrass::ConnectorDefinitionVersion`
- `AWS::Greengrass::CoreDefinition`
- `AWS::Greengrass::CoreDefinitionVersion`
- `AWS::Greengrass::DeviceDefinition`
- `AWS::Greengrass::DeviceDefinitionVersion`
- `AWS::Greengrass::FunctionDefinition`
- `AWS::Greengrass::FunctionDefinitionVersion`
- `AWS::Greengrass::Group`
- `AWS::Greengrass::GroupVersion`
- `AWS::Greengrass::LoggerDefinition`
- `AWS::Greengrass::LoggerDefinitionVersion`
- `AWS::Greengrass::ResourceDefinition`
- `AWS::Greengrass::ResourceDefinitionVersion`
- `AWS::Greengrass::SubscriptionDefinition`
- `AWS::Greengrass::SubscriptionDefinitionVersion`
- `AWS::IAM::AccessKey`
- `AWS::IAM::UserToGroupAddition`
- `AWS::IoT::PolicyPrincipalAttachment`
- `AWS::IoT::ThingPrincipalAttachment`
- `AWS::IoTThingsGraph::FlowTemplate`
- `AWS::KinesisAnalytics::Application`
- `AWS::KinesisAnalytics::ApplicationOutput`
- `AWS::KinesisAnalytics::ApplicationReferenceDataSource`
- `AWS::KinesisAnalyticsV2::ApplicationCloudWatchLoggingOption`
- `AWS::KinesisAnalyticsV2::ApplicationOutput`
- `AWS::KinesisAnalyticsV2::ApplicationReferenceDataSource`
- `AWS::LakeFormation::DataLakeSettings`
- `AWS::LakeFormation::Permissions`
- `AWS::LakeFormation::Resource`

- `AWS::ManagedBlockchain::Member`
- `AWS::ManagedBlockchain::Node`
- `AWS::MediaConvert::JobTemplate`
- `AWS::MediaConvert::Preset`
- `AWS::MediaConvert::Queue`
- `AWS::MediaLive::Channel`
- `AWS::MediaLive::Input`
- `AWS::MediaLive::InputSecurityGroup`
- `AWS::MediaStore::Container`
- `AWS::OpsWorks::App`
- `AWS::OpsWorks::ElasticLoadBalancerAttachment`
- `AWS::OpsWorks::Instance`
- `AWS::OpsWorks::Layer`
- `AWS::OpsWorks::Stack`
- `AWS::OpsWorks::UserProfile`
- `AWS::OpsWorks::Volume`
- `AWS::Pinpoint::ADMChannel`
- `AWS::Pinpoint::APNSChannel`
- `AWS::Pinpoint::APNSSandboxChannel`
- `AWS::Pinpoint::APNSVoipChannel`
- `AWS::Pinpoint::APNSVoipSandboxChannel`
- `AWS::Pinpoint::App`
- `AWS::Pinpoint::ApplicationSettings`
- `AWS::Pinpoint::BaiduChannel`
- `AWS::Pinpoint::Campaign`
- `AWS::Pinpoint::EmailChannel`
- `AWS::Pinpoint::EmailTemplate`
- `AWS::Pinpoint::EventStream`
- `AWS::Pinpoint::GCMChannel`

- `AWS::Pinpoint::PushTemplate`
- `AWS::Pinpoint::Segment`
- `AWS::Pinpoint::SMSChannel`
- `AWS::Pinpoint::SmsTemplate`
- `AWS::Pinpoint::VoiceChannel`
- `AWS::PinpointEmail::ConfigurationSet`
- `AWS::PinpointEmail::ConfigurationSetEventDestination`
- `AWS::PinpointEmail::DedicatedIpPool`
- `AWS::PinpointEmail::Identity`
- `AWS::QLDB::Ledger`
- `AWS::RDS::DBSecurityGroup`
- `AWS::RDS::DBSecurityGroupIngress`
- `AWS::Redshift::ClusterSecurityGroup`
- `AWS::Redshift::ClusterSecurityGroupIngress`
- `AWS::Route53::RecordSet`
- `AWS::Route53::RecordSetGroup`
- `AWS::SageMaker::CodeRepository`
- `AWS::SageMaker::EndpointConfig`
- `AWS::SageMaker::Model`
- `AWS::SageMaker::NotebookInstance`
- `AWS::SageMaker::NotebookInstanceLifecycleConfig`
- `AWS::SageMaker::Workteam`
- `AWS::SDB::Domain`
- `AWS::ServiceCatalog::AcceptedPortfolioShare`
- `AWS::ServiceCatalog::LaunchRoleConstraint`
- `AWS::ServiceCatalog::Portfolio`
- `AWS::ServiceCatalog::StackSetConstraint`
- `AWS::ServiceDiscovery::HttpNamespace`
- `AWS::ServiceDiscovery::Instance`

- `AWS::ServiceDiscovery::PrivateDnsNamespace`
- `AWS::ServiceDiscovery::PublicDnsNamespace`
- `AWS::ServiceDiscovery::Service`
- `AWS::SES::ReceiptFilter`
- `AWS::SES::ReceiptRule`
- `AWS::SES::ReceiptRuleSet`
- `AWS::SSM::MaintenanceWindow`
- `AWS::SSM::MaintenanceWindowTarget`
- `AWS::SSM::MaintenanceWindowTask`
- `AWS::WAF::ByteMatchSet`
- `AWS::WAF::IPSet`
- `AWS::WAF::Rule`
- `AWS::WAF::SizeConstraintSet`
- `AWS::WAF::SqlInjectionMatchSet`
- `AWS::WAF::WebACL`
- `AWS::WAF::XssMatchSet`
- `AWS::WAFRegional::ByteMatchSet`
- `AWS::WAFRegional::GeoMatchSet`
- `AWS::WAFRegional::IPSet`
- `AWS::WAFRegional::RateBasedRule`
- `AWS::WAFRegional::RegexPatternSet`
- `AWS::WAFRegional::Rule`
- `AWS::WAFRegional::SizeConstraintSet`
- `AWS::WAFRegional::SqlInjectionMatchSet`
- `AWS::WAFRegional::WebACL`
- `AWS::WAFRegional::WebACLAssociation`
- `AWS::WAFRegional::XssMatchSet`
- `AWS::WorkSpaces::Workspace`

- `AWS::AmazonMQ::ConfigurationAssociation`
- `AWS::ApiGateway::DomainNameAccessAssociation`
- `AWS::AppConfig::ExtensionAssociation`
- `AWS::AppStream::ApplicationEntitlementAssociation`
- `AWS::AppStream::ApplicationFleetAssociation`
- `AWS::AppSync::DomainNameApiAssociation`
- `AWS::AppSync::SourceApiAssociation`
- `AWS::CleanRooms::ConfiguredTableAssociation`
- `AWS::CleanRooms::IdNamespaceAssociation`
- `AWS::CodeGuruReviewer::RepositoryAssociation`
- `AWS::Cognito::IdentityPoolRoleAttachment`
- `AWS::Cognito::UserPoolRiskConfigurationAttachment`
- `AWS::Cognito::UserPoolUICustomizationAttachment`
- `AWS::Cognito::UserPoolUserToGroupAttachment`
- `AWS::Connect::IntegrationAssociation`
- `AWS::Deadline::QueueFleetAssociation`
- `AWS::Deadline::QueueLimitAssociation`
- `AWS::EC2::EIPAssociation`
- `AWS::EC2::EnclaveCertificateIamRoleAssociation`
- `AWS::EC2::GatewayRouteTableAssociation`
- `AWS::EC2::IPAMResourceDiscoveryAssociation`
- `AWS::EC2::IpPoolRouteTableAssociation`
- `AWS::EC2::LocalGatewayRouteTableVPCAssociation`
- `AWS::EC2::LocalGatewayRouteTableVirtualInterfaceGroupAssociation`
- `AWS::EC2::NetworkInterfaceAttachment`
- `AWS::EC2::RouteServerAssociation`
- `AWS::EC2::SecurityGroupVpcAssociation`
- `AWS::EC2::SubnetNetworkAclAssociation`

- `AWS::EC2::SubnetRouteTableAssociation`
- `AWS::EC2::TransitGatewayAttachment`
- `AWS::EC2::TransitGatewayMulticastDomainAssociation`
- `AWS::EC2::TransitGatewayPeeringAttachment`
- `AWS::EC2::TransitGatewayRouteTableAssociation`
- `AWS::EC2::TransitGatewayVpcAttachment`
- `AWS::EC2::VPCDHCPOptionsAssociation`
- `AWS::EC2::VPCGatewayAttachment`
- `AWS::EC2::VolumeAttachment`
- `AWS::ECS::ClusterCapacityProviderAssociations`
- `AWS::EKS::PodIdentityAssociation`
- `AWS::FSx::DataRepositoryAssociation`
- `AWS::FSx::S3AccessPointAttachment`
- `AWS::GlobalAccelerator::CrossAccountAttachment`
- `AWS::LakeFormation::TagAssociation`
- `AWS::NetworkFirewall::VpcEndpointAssociation`
- `AWS::NetworkManager::ConnectAttachment`
- `AWS::NetworkManager::CustomerGatewayAssociation`
- `AWS::NetworkManager::DirectConnectGatewayAttachment`
- `AWS::NetworkManager::LinkAssociation`
- `AWS::NetworkManager::SiteToSiteVpnAttachment`
- `AWS::NetworkManager::TransitGatewayRouteTableAttachment`
- `AWS::NetworkManager::VpcAttachment`
- `AWS::Notifications::ChannelAssociation`
- `AWS::Notifications::ManagedNotificationAccountContactAssociation`
- `AWS::Notifications::ManagedNotificationAdditionalChannelAssociation`
- `AWS::Notifications::OrganizationalUnitAssociation`
- `AWS::ResourceExplorer2::DefaultViewAssociation`

- `AWS::Route53Profiles::ProfileAssociation`
- `AWS::Route53Profiles::ProfileResourceAssociation`
- `AWS::Route53Resolver::FirewallRuleGroupAssociation`
- `AWS::Route53Resolver::ResolverQueryLoggingConfigAssociation`
- `AWS::Route53Resolver::ResolverRuleAssociation`
- `AWS::SSM::Association`
- `AWS::SecretsManager::SecretTargetAttachment`
- `AWS::SecurityHub::PolicyAssociation`
- `AWS::ServiceCatalog::PortfolioPrincipalAssociation`
- `AWS::ServiceCatalog::PortfolioProductAssociation`
- `AWS::ServiceCatalog::ServiceActionAssociation`
- `AWS::ServiceCatalog::TagOptionAssociation`
- `AWS::ServiceCatalogAppRegistry::AttributeGroupAssociation`
- `AWS::ServiceCatalogAppRegistry::ResourceAssociation`
- `AWS::VpcLattice::ServiceNetworkResourceAssociation`
- `AWS::VpcLattice::ServiceNetworkServiceAssociation`
- `AWS::VpcLattice::ServiceNetworkVpcAssociation`
- `AWS::WAFv2::WebACLAssociation`
- `AWS::Wisdom::AssistantAssociation`
- `AWS::WorkspacesInstances::VolumeAssociation`
- `AWS::IAM::Policy`
- `AWS::SNS::TopicPolicy`
- `AWS::SQS::QueuePolicy`
- `AWS::EC2::NetworkAclEntry`
- `AWS::EC2::VPNGatewayRoutePropagation`
- `AWS::CloudFormation::Stack`
- `AWS::CloudWatch::MetricStream`
- `AWS::WorkSpaces::ConnectionAlias`

- `AWS::IoT::ProvisioningTemplate`
- `AWS::MediaPackage::Channel`
- `AWS::CloudFront::OriginRequestPolicy`
- `AWS::Route53Resolver::ResolverQueryLoggingConfig`
- `AWS::NetworkManager::TransitGatewayRegistration`
- `AWS::ImageBuilder::Image`
- `AWS::Config::ConformancePack`
- `AWS::S3::AccessPoint`
- `AWS::CodeStarConnections::Connection`
- `AWS::CloudFront::CachePolicy`
- `AWS::FMS::NotificationChannel`
- `AWS::ImageBuilder::InfrastructureConfiguration`
- `AWS::Detective::Graph`
- `AWS::EC2::CarrierGateway`
- `AWS::CloudWatch::CompositeAlarm`
- `AWS::CodeArtifact::Repository`
- `AWS::GroundStation::DataflowEndpointGroup`
- `AWS::ElasticLoadBalancingV2::Listener`
- `AWS::ImageBuilder::ImageRecipe`
- `AWS::NetworkManager::Device`
- `AWS::Kendra::DataSource`
- `AWS::Timestream::Database`
- `AWS::CodeGuruProfiler::ProfilingGroup`
- `AWS::Lambda::EventSourceMapping`
- `AWS::ECR::Repository`
- `AWS::WAFv2::IPSet`
- `AWS::GameLift::Alias`
- `AWS::IoTSiteWise::Asset`

- `AWS::OpsWorksCM::Server`
- `AWS::IoT::Authorizer`
- `AWS::WAFv2::RuleGroup`
- `AWS::NetworkManager::Site`
- `AWS::ResourceGroups::Group`
- `AWS::MediaPackage::PackagingConfiguration`
- `AWS::ImageBuilder::ImagePipeline`
- `AWS::ECS::TaskDefinition`
- `AWS::Macie::CustomDataIdentifier`
- `AWS::MediaPackage::OriginEndpoint`
- `AWS::Logs::LogGroup`
- `AWS::CodeArtifact::Domain`
- `AWS::Kendra::Faq`
- `AWS::ECS::TaskSet`
- `AWS::WAFv2::RegexPatternSet`
- `AWS::ECS::Cluster`
- `AWS::SSO::Assignment`
- `AWS::GlobalAccelerator::Listener`
- `AWS::ServiceCatalog::CloudFormationProvisionedProduct`
- `AWS::RDS::DBProxy`
- `AWS::EC2::FlowLog`
- `AWS::ImageBuilder::Component`
- `AWS::CloudFront::RealtimeLogConfig`
- `AWS::NetworkManager::GlobalNetwork`
- `AWS::RDS::DBProxyTargetGroup`
- `AWS::WAFv2::WebACL`
- `AWS::IVS::StreamKey`
- `AWS::IVS::PlaybackKeyPair`
- `AWS::Macie::Session`

- `AWS::Route53::HealthCheck`
- `AWS::Synthetics::Canary`
- `AWS::Lambda::CodeSigningConfig`
- `AWS::EFS::AccessPoint`
- `AWS::Timestream::Table`
- `AWS::MediaPackage::PackagingGroup`
- `AWS::ECS::PrimaryTaskSet`
- `AWS::Config::ConfigurationAggregator`
- `AWS::GroundStation::Config`
- `AWS::IoTSiteWise::AssetModel`
- `AWS::SES::ConfigurationSet`
- `AWS::ImageBuilder::DistributionConfiguration`
- `AWS::Config::OrganizationConformancePack`
- `AWS::EC2::LocalGatewayRoute`
- `AWS::KMS::Key`
- `AWS::Detective::MemberInvitation`
- `AWS::EKS::FargateProfile`
- `AWS::MediaPackage::Asset`
- `AWS::GlobalAccelerator::EndpointGroup`
- `AWS::Macie::FindingsFilter`
- `AWS::IoT::Certificate`
- `AWS::SageMaker::MonitoringSchedule`
- `AWS::IVS::Channel`
- `AWS::Kendra::Index`
- `AWS::EventSchemas::RegistryPolicy`
- `AWS::KinesisFirehose::DeliveryStream`
- `AWS::GlobalAccelerator::Accelerator`
- `AWS::EC2::PrefixList`
- `AWS::GameLift::GameServerGroup`

- `AWS::NetworkManager::Link`
- `AWS::EFS::FileSystem`
- `AWS::Route53::HostedZone`
- `AWS::GroundStation::MissionProfile`
- `AWS::KMS::Alias`
- `AWS::FMS::Policy`
- `AWS::SSO::PermissionSet`
- `AWS::StepFunctions::StateMachine`
- `AWS::QLDB::Stream`
- `AWS::IoTSiteWise::Gateway`
- `AWS::ECS::Service`
- `AWS::ECS::CapacityProvider`
- `AWS::EC2::SecurityGroup`
- `AWS::EC2::SecurityGroupIngress`
- `AWS::EC2::SecurityGroupEgress`
- `AWS::EC2::EC2Fleet`
- `AWS::IAM::Group`
- `AWS::IAM::Role`
- `AWS::IAM::User`
- `AWS::ApiGateway::GatewayResponse`
- `AWS::S3::BucketPolicy`
- `AWS::SNS::Topic`
- `AWS::SNS::Subscription`
- `AWS::RDS::DBInstance`
- `AWS::RDS::DBParameterGroup`
- `AWS::RDS::DBCluster`
- `AWS::RDS::DBClusterParameterGroup`
- `AWS::RDS::DBSubnetGroup`
- `AWS::RDS::EventSubscription`

- `AWS::RDS::GlobalCluster`
- `AWS::RDS::OptionGroup`
- `AWS::Neptune::DBInstance`
- `AWS::Neptune::DBParameterGroup`
- `AWS::Neptune::DBCluster`
- `AWS::Neptune::DBClusterParameterGroup`
- `AWS::Neptune::DBSubnetGroup`
- `AWS::Redshift::Cluster`
- `AWS::Redshift::ClusterParameterGroup`
- `AWS::Redshift::ClusterSubnetGroup`
- `AWS::Redshift::EndpointAccess`
- `AWS::Redshift::EndpointAuthorization`
- `AWS::Redshift::EventSubscription`
- `AWS::Redshift::ScheduledAction`
- `AWS::ElastiCache::SubnetGroup`

Actualizar pilas directamente

Cuando desee implementar rápidamente actualizaciones directas en la pila, realice una actualización directa. Con una actualización directa, envía una plantilla o parámetros de entrada que especifican actualizaciones de los recursos en la pila y CloudFormation los implementa inmediatamente. Si desea utilizar una plantilla para realizar las actualizaciones, puede modificar la plantilla actual y almacenarla localmente o en un bucket de Amazon S3.

Para las propiedades de recursos que no admiten actualizaciones, debe mantener los valores actuales. Para obtener una vista previa de los cambios que CloudFormation realizará en la pila antes de actualizarlo, utilice conjuntos de cambios. Para obtener más información, consulte [Actualización de pilas de CloudFormation con conjuntos de cambios](#).

Al actualizar una pila, CloudFormation podría interrumpir los recursos o sustituir los recursos actualizados, en función de las propiedades que actualice. Para obtener más información acerca de las conductas de actualización, consulte [Comprensión de los comportamientos de actualización de los recursos de la pila](#).

Para actualizar una pila de (consola)

1. Inicie sesión en la Consola de administración de AWS y abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation>.
2. En la barra de navegación de la parte superior de la pantalla, elija la Región de AWS.
3. En la página Pilas, seleccione la pila en ejecución que desea actualizar.
4. En el panel de detalles de la pila, elija Update (Actualizar).
5. Si no ha modificado la plantilla de pilas, seleccione Usar plantilla existente y, a continuación, seleccione Siguiente.

Si ha modificado la plantilla, seleccione Sustituir plantilla existente y especifique la ubicación de la plantilla actualizada en la sección Especificar plantilla:

- Para una plantilla almacenada localmente en su equipo, seleccione Upload a template file (Cargar un archivo de plantilla). Elija Choose File (Elegir archivo) para navegar hasta el archivo, selecciónelo y, a continuación, elija Next (Siguiente).

Note

Si carga un archivo de plantilla local, CloudFormation lo carga a un bucket de Amazon Simple Storage Service (Amazon S3) en su Cuenta de AWS. Si todavía no tiene un bucket de S3 creado con CloudFormation, crea un bucket único para cada región en la que carga un archivo de plantilla. Si ya tiene un bucket de S3 creado por CloudFormation en su Cuenta de AWS, CloudFormation agrega la plantilla a ese bucket.

Consideraciones que deben tenerse en cuenta acerca de los buckets de S3 creados con CloudFormation

- Cualquiera con permisos de Amazon S3 en su Cuenta de AWS puede acceder a los buckets.
- CloudFormation crea los buckets con el cifrado del lado del servidor habilitado de forma predeterminada; por lo tanto, cifra todos los objetos almacenados en el bucket.

Puede administrar directamente las opciones de cifrado para los buckets creados con CloudFormation; por ejemplo, mediante la consola de Amazon S3 en <https://console.aws.amazon.com/s3/> o la AWS CLI. Para obtener más información, consulte [Establecer el comportamiento del cifrado predeterminado del lado del](#)

[servidor para los buckets de Amazon S3](#), en la Guía del usuario de Amazon Simple Storage Service.

- Puede utilizar su propio bucket y administrar sus permisos cargando manualmente las plantillas a Amazon S3. Al crear o actualizar una pila, especifique la URL de Amazon S3 de un archivo de plantilla.
- En el caso de una plantilla almacenada en un bucket de Amazon S3, seleccione Amazon S3 URL (URL de Amazon S3). Escriba o pegue la URL de la plantilla y luego seleccione Next (Siguiente).

Si tiene una plantilla en un bucket compatible con el control de versiones, puede especificar una versión concreta de la plantilla al adjuntar `?versionId=version-id` a la URL. Para obtener más información, consulte [Trabajo con objetos en un bucket con control de versiones habilitado](#) en la Guía del usuario de la consola de Amazon Simple Storage Service.

Si se detecta algún problema de sintaxis, la consola brinda mensajes de error que ayudan a corregir la plantilla.

6. Si la plantilla contiene parámetros, en la página Specify stack details (Especificar detalles de pila) escriba o modifique los valores de los parámetros y, a continuación, elija Next (Siguiente).

CloudFormation rellena cada parámetro con el valor que actualmente está establecido en la pila, a excepción de los parámetros declarados con el atributo NoEcho; no obstante, puede seguir utilizando los valores actuales si elige Usar valor existente.

Para obtener más información sobre el uso de NoEcho para enmascarar información confidencial además del uso de parámetros dinámicos para la administración de secretos, consulte la práctica recomendada de [No integre credenciales en sus plantillas](#).

7. En la página Configure stack options (Configurar opciones de pila), puede actualizar las etiquetas y los permisos aplicados a la pila. También puede modificar las opciones avanzadas como la política de la pila o la configuración de restauración, así como actualizar el tema de notificación de Amazon SNS. Para obtener más información sobre estas opciones, consulte [Configurar las opciones de la pila](#).
8. Si la plantilla contiene recursos de IAM, en Capacidades elija Confirmando que esta plantilla pueda crear recursos de IAM para especificar que desea utilizar recursos de IAM en la plantilla. Para obtener más información, consulte [Reconocimiento de recursos de IAM en plantillas de CloudFormation](#).

9. Elija Siguiente para continuar.
10. Revise la información de la pila y los cambios que ha enviado.

Compruebe que ha enviado la información correcta, como la URL de la plantilla o los valores de los parámetros correctos.

En la sección Vista previa de los cambios, compruebe que CloudFormation realice todos los cambios que espera. Por ejemplo, puede comprobar que CloudFormation agregue, elimine y modifique los recursos que deseaba agregar, eliminar o modificar. CloudFormation genera esta vista previa mediante la creación de un conjunto de cambios para la pila. Para obtener más información, consulte [Actualización de pilas de CloudFormation con conjuntos de cambios](#).

11. Cuando esté satisfecho con los cambios, elija Update stack (Actualizar pila).

 Note

En este momento, también tiene la opción de ver el conjunto de cambios para revisar las actualizaciones propuestas más detenidamente. Para hacerlo, elija View change set (Ver conjunto de cambios) en lugar de Update stack (Actualizar pila). CloudFormation muestra el conjunto de cambios generados en función de sus actualizaciones. Cuando esté listo para actualizar la pila, elija Execute (Ejecutar).

CloudFormation muestra la página de detalles de la pila, con el panel Events (Eventos) seleccionado. La pila ahora tiene un estado UPDATE_IN_PROGRESS. Una vez que CloudFormation haya terminado correctamente de actualizar la pila, establece el estado de la pila en UPDATE_COMPLETE.

Si se produce un error en la actualización de la pila, CloudFormation restaura automáticamente los cambios y establece el estado de la pila en UPDATE_ROLLBACK_COMPLETE.

 Note

Puede cancelar una actualización mientras se está en el estado UPDATE_IN_PROGRESS. Para obtener más información, consulte [Cancelar una actualización de pila](#).

Actualización de una pila con la línea de comandos

Puede utilizar uno de los siguientes comandos:

- [update-stack](#) (AWS CLI)
- [Update-CFNStack](#) (AWS Tools for Windows PowerShell)

Para ver ejemplos del uso de la línea de comandos para actualizar una pila, consulte [Ejemplos de comandos de operación de pila de CloudFormation para la AWS CLI y PowerShell](#).

Cancelar una actualización de pila

Una vez ha comenzado la actualización de una pila, puede cancelarla si la pila sigue en el estado UPDATE_IN_PROGRESS. Una vez finalizada la actualización, no podrá cancelarla. No obstante, puede volver a actualizar una pila con una configuración anterior.

Si cancela una actualización de pila, esta se revierte a la configuración de pila que existía antes de iniciar la actualización de pila.

Temas

- [Cancelación de una actualización de una pila de \(consola\)](#)
- [Cancelación de una actualización de una pila \(AWS CLI\)](#)

Cancelación de una actualización de una pila de (consola)

1. Inicie sesión en la Consola de administración de AWS y abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation>.
2. En la barra de navegación de la parte superior de la pantalla, elija la Región de AWS en donde se encuentra la pila.
3. Desde la página Pilas, elija la pila que se esté actualizando actualmente. El estado debe ser UPDATE_IN_PROGRESS.
4. Seleccione Stack actions (Acciones de pila) y, luego, Cancel update stack (Cancelar actualización de pila).
5. Para continuar con la cancelación de la actualización, elija Cancel update (Cancelar actualización). De lo contrario, elija Cancel (Cancelar) para reanudar la actualización.

La pila avanza al estado `UPDATE_ROLLBACK_IN_PROGRESS`. Una vez completada la cancelación de la actualización, se establece la pila como `UPDATE_ROLLBACK_COMPLETE`.

Cancelación de una actualización de una pila (AWS CLI)

Utilice el comando [cancel-update-stack](#) para cancelar una actualización. Para obtener más información, consulte [Cancelar una actualización de pila](#).

Eliminar una pila de la consola de CloudFormation

Si ya no necesita los recursos de una pila, puede eliminarlos por completo.

Al eliminar una pila, CloudFormation elimina todos los recursos de esa pila, a menos que haya utilizado una política de eliminación para retener recursos específicos. Para obtener más información, consulte [DeletionPolicy atributo](#).

Eliminación de una pila (consola)

1. Inicie sesión en la Consola de administración de AWS y abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation>.
2. En la barra de navegación de la parte superior de la pantalla, elija la Región de AWS en donde se encuentra la pila.
3. En la página Pilas, seleccione la pila que desea eliminar. La pila se debe estar ejecutando en este momento.
4. Elija Eliminar.
5. Cuando se le pida confirmación, seleccione Eliminar.

Note

La operación de eliminación de la pila no se puede detener una vez que comienza. La pila avanza al estado `DELETE_IN_PROGRESS`.

Una vez que se haya completado la eliminación de la pila, la pila estará en el estado `DELETE_COMPLETE`. Las pilas en el estado `DELETE_COMPLETE` no se muestran en la consola de CloudFormation de forma predeterminada. Para visualizar las pilas eliminadas, debe cambiar el filtro de visualización de la pila tal y como se describe en [Visualización de las pilas eliminadas desde la consola de CloudFormation](#).

Cómo forzar la eliminación de una pila (consola)

La eliminación de una pila podría fallar si un recurso dentro de la pila no se elimina correctamente. Por ejemplo, CloudFormation no podrá eliminar un recurso del que también dependa otra pila. Cualquier recurso que no se haya eliminado permanecerá hasta que pueda eliminar correctamente la pila. Si la eliminación falla y devuelve un estado `DELETE_FAILED`, puede optar por intentarlo nuevamente mediante uno de dos métodos.

1. En la página Pilas de la consola de CloudFormation, elija la pila para la que desea forzar la eliminación.
2. En el panel de detalles de la pila, seleccione Reintentar eliminación.
3. Elija entre las opciones siguientes:
 - Eliminar esta pila, pero retener los recursos: esta opción le permite seleccionar los recursos específicos que originalmente no se pudieron eliminar, pero que desea retener durante la eliminación forzada de la pila.
 - Forzar la eliminación de toda la pila: esta opción retiene todos los recursos que no se pudieron eliminar y retiene sus dependencias.
4. Elija Eliminar para iniciar el proceso de eliminación forzada con sus selecciones.

Revisión de los recursos retenidos (consola)

Después de eliminar la pila, puede ver los recursos retenidos en la consola.

1. En la lista de pilas, seleccione la opción Filtrar por estado y elija Eliminadas.
2. Seleccione la pila eliminada.
3. Elija la pestaña Recursos.
4. Todos los recursos retenidos muestran el estado `DELETE_SKIPPED`.
5. Elija el recurso retenido que desea revisar.

Eliminación de una pila con la línea de comandos

Puede utilizar uno de los siguientes comandos:

- [delete-stack](#) (AWS CLI)
- [Remove-CFNStack](#) (AWS Tools for Windows PowerShell)

Para ver ejemplos del uso de la línea de comandos para eliminar una pila, consulte [Ejemplos de comandos de operación de pila de CloudFormation para la AWS CLI y PowerShell](#).

Recursos relacionados

Para obtener ayuda para solucionar los errores de eliminación de pila, consulte el tema de resolución de problemas [Errores al eliminar la pila](#).

Para obtener más información acerca de cómo proteger las pilas para que no se eliminen accidentalmente, consulte [Protección de las pilas de CloudFormation para evitar la eliminación](#).

Visualización de las pilas eliminadas desde la consola de CloudFormation

De forma predeterminada, la consola de CloudFormation no muestra pilas con un estado DELETE_COMPLETE. Para visualizar información sobre las pilas eliminadas, debe cambiar la visualización de la pila.

Visualización de pilas eliminadas

1. Inicie sesión en la Consola de administración de AWS y abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation>.
2. En la barra de navegación de la parte superior de la pantalla, elija la Región de AWS en donde se encuentra la pila eliminada.
3. En la página Pilas, seleccione Eliminado en el menú desplegable del Estado del filtro.

CloudFormation enumera todas las pilas con el estado DELETE_COMPLETE.

Véase también

- [Eliminar una pila de la consola de CloudFormation](#)
- [Visualización de la información de la pila desde la consola de CloudFormation](#)

Monitorizar el progreso de la pila

En esta sección se describe cómo supervisar una implementación de pila que se encuentra actualmente en curso. CloudFormation proporciona una lista cronológica detallada de los eventos de implementación, que muestra el progreso y los problemas encontrados durante la implementación.

Temas

- [Visualización de eventos de pilas de CloudFormation](#)
- [Visualización de eventos de la pila por operación](#)
- [Visualización de un cronograma de la implementación de una pila de CloudFormation](#)
- [Comprensión de los eventos de creación de la pila de CloudFormation](#)
- [Monitorización del progreso de una actualización de pila](#)
- [Continuación de la restauración de una actualización](#)
- [Determine la causa de una falla en la pila](#)
- [Elija cómo gestionar los errores al aprovisionar recursos](#)

Visualización de eventos de pilas de CloudFormation

Puede ver los eventos de pilas para monitorear el avance y el estado de la pila y los recursos de la pila. Los eventos de pilas ayudan a comprender cuándo se crean, actualizan o eliminan los recursos y si la implementación de pilas se está desarrollando según lo previsto.

Temas

- [Ver eventos de pila \(consola\)](#)
- [Ver los eventos de la pila \(AWS CLI\)](#)
- [Códigos de estado de pilas](#)

Ver eventos de pila (consola)

Para ver eventos de pila

1. Abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation>.
2. En la barra de navegación de la parte superior de la pantalla, elija la Región de AWS en la que creó la pila.
3. En la página Stacks (Pilas) de la consola de CloudFormation, seleccione el nombre de la pila. CloudFormation muestra la información de la pila que haya seleccionado.
4. Seleccione la pestaña Eventos para ver los eventos de la pila que CloudFormation generó para la pila.

CloudFormation actualiza automáticamente los eventos de la pila cada minuto. Además, CloudFormation muestra la insignia Nuevos eventos disponibles cuando se producen nuevos eventos de la pila. Seleccione el icono de actualización para cargar estos eventos en la lista. Al visualizar eventos de creación de la pila, puede comprender la secuencia de eventos que conducen a la creación de la pila (o error, si está depurando la pila).

Mientras se crea, la pila aparecerá en la página Pilas con el estado `CREATE_IN_PROGRESS`. Una vez que se cree correctamente, el estado de la pila cambia a `CREATE_COMPLETE`.

Para obtener más información, consulte [Comprensión de los eventos de creación de la pila de CloudFormation](#) y [Monitorización del progreso de una actualización de pila](#).

Ver los eventos de la pila (AWS CLI)

Alternativamente, puede usar el comando [describe-stack-events](#) mientras se está creando la pila para ver eventos a medida que se notifican.

El siguiente comando `describe-stack-events` describe los eventos de la pila *my-stack*.

```
aws cloudformation describe-stack-events --stack-name my-stack
```

A continuación, se muestra un ejemplo de respuesta.

```
{
  "StackEvents": [
    {
      "StackId": "arn:aws:cloudformation:aws-region:123456789012:stack/my-stack/64726230-7edf-11f0-8a36-06453a64f325",
      "EventId": "7b755820-7edf-11f0-ab15-0673b09f3847",
      "StackName": "my-stack",
      "LogicalResourceId": "my-stack",
      "PhysicalResourceId": "arn:aws:cloudformation:aws-region:123456789012:stack/my-stack/64726230-7edf-11f0-8a36-06453a64f325",
      "ResourceType": "AWS::CloudFormation::Stack",
      "Timestamp": "2025-08-21T22:37:56.243000+00:00",
      "ResourceStatus": "CREATE_COMPLETE",
      "ClientRequestToken": "token"
    },
    {
      "StackId": "arn:aws:cloudformation:aws-region:123456789012:stack/my-stack/64726230-7edf-11f0-8a36-06453a64f325",
      "EventId": "WebServer-CREATE_COMPLETE-2025-08-21T22:37:54.356Z",
```

```

        "StackName": "my-stack",
        "LogicalResourceId": "WebServer",
        "PhysicalResourceId": "i-099df76cb31b866a9",
        "ResourceType": "AWS::EC2::Instance",
        "Timestamp": "2025-08-21T22:37:54.356000+00:00",
        "ResourceStatus": "CREATE_COMPLETE",
        "ResourceProperties": "{\"UserData\":
\\\"IyEvYmluL2Jhc2gKeXVtIGluc3RhbGwgLXkgYXdzLWNmbi1ib290c3RyYXAKL29wdC9hd3MvYmluL2Nmbi1pbml0IC12I
\\\",\\\"ImageId\\\":\\\"ami-0bbc328167dee8f3c\\\",\\\"InstanceType\\\":\\\"t2.micro\\\",
\\\"SecurityGroupIds\\\":[\\\"my-stack-WebServerSecurityGroup-n8A43bQT1ty2\\\"],\\\"Tags\\\":
[\\\"Value\\\":\\\"Bootstrap Tutorial Web Server\\\",\\\"Key\\\":\\\"Name\\\"]}]\",
        \"ClientRequestToken\": \"token\"
    },
    {
        \"StackId\": \"arn:aws:cloudformation:aws-region:123456789012:stack/my-
stack/64726230-7edf-11f0-8a36-06453a64f325\",
        \"EventId\": \"WebServer-CREATE_IN_PROGRESS-2025-08-21T22:37:31.226Z\",
        \"StackName\": \"my-stack\",
        \"LogicalResourceId\": \"WebServer\",
        \"PhysicalResourceId\": \"i-099df76cb31b866a9\",
        \"ResourceType\": \"AWS::EC2::Instance\",
        \"Timestamp\": \"2025-08-21T22:37:31.226000+00:00\",
        \"ResourceStatus\": \"CREATE_IN_PROGRESS\",
        \"ResourceStatusReason\": \"Resource creation Initiated\",
        \"ResourceProperties\": \"{\\\"UserData\\\":
\\\"IyEvYmluL2Jhc2gKeXVtIGluc3RhbGwgLXkgYXdzLWNmbi1ib290c3RyYXAKL29wdC9hd3MvYmluL2Nmbi1pbml0IC12I
\\\",\\\"ImageId\\\":\\\"ami-0bbc328167dee8f3c\\\",\\\"InstanceType\\\":\\\"t2.micro\\\",
\\\"SecurityGroupIds\\\":[\\\"my-stack-WebServerSecurityGroup-n8A43bQT1ty2\\\"],\\\"Tags\\\":
[\\\"Value\\\":\\\"Bootstrap Tutorial Web Server\\\",\\\"Key\\\":\\\"Name\\\"]}]\",
        \"ClientRequestToken\": \"token\"
    },
    {
        \"StackId\": \"arn:aws:cloudformation:aws-region:123456789012:stack/my-
stack/64726230-7edf-11f0-8a36-06453a64f325\",
        \"EventId\": \"WebServer-CREATE_IN_PROGRESS-2025-08-21T22:37:29.210Z\",
        \"StackName\": \"my-stack\",
        \"LogicalResourceId\": \"WebServer\",
        \"PhysicalResourceId\": \"\",
        \"ResourceType\": \"AWS::EC2::Instance\",
        \"Timestamp\": \"2025-08-21T22:37:29.210000+00:00\",
        \"ResourceStatus\": \"CREATE_IN_PROGRESS\",
        \"ResourceProperties\": \"{\\\"UserData\\\":
\\\"IyEvYmluL2Jhc2gKeXVtIGluc3RhbGwgLXkgYXdzLWNmbi1ib290c3RyYXAKL29wdC9hd3MvYmluL2Nmbi1pbml0IC12I
\\\",\\\"ImageId\\\":\\\"ami-0bbc328167dee8f3c\\\",\\\"InstanceType\\\":\\\"t2.micro\\\",

```

```

\"SecurityGroupIds\": [\"my-stack-WebServerSecurityGroup-n8A43bQT1ty2\"], \"Tags\":
[{\\"Value\\\": \\"Bootstrap Tutorial Web Server\\\", \\"Key\\\": \\"Name\\\"}]]\",
  \"ClientRequestToken\": \"token\"
},
{
  \"StackId\": \"arn:aws:cloudformation:aws-region:123456789012:stack/my-
stack/64726230-7edf-11f0-8a36-06453a64f325\",
  \"EventId\": \"WebServerSecurityGroup-
CREATE_COMPLETE-2025-08-21T22:37:28.803Z\",
  \"StackName\": \"my-stack\",
  \"LogicalResourceId\": \"WebServerSecurityGroup\",
  \"PhysicalResourceId\": \"my-stack-WebServerSecurityGroup-n8A43bQT1ty2\",
  \"ResourceType\": \"AWS::EC2::SecurityGroup\",
  \"Timestamp\": \"2025-08-21T22:37:28.803000+00:00\",
  \"ResourceStatus\": \"CREATE_COMPLETE\",
  \"ResourceProperties\": \"{\\\"GroupDescription\\\": \\"Allow HTTP access from my IP
address\\\", \\"SecurityGroupIngress\\\": [{\\\"CidrIp\\\": \\"0.0.0.0/0\\\", \\"Description\\\": \\"HTTP
\\\", \\"FromPort\\\": \\"80\\\", \\"ToPort\\\": \\"80\\\", \\"IpProtocol\\\": \\"tcp\\\"}]]}\",
  \"ClientRequestToken\": \"token\"
},
{
  \"StackId\": \"arn:aws:cloudformation:aws-region:123456789012:stack/my-
stack/64726230-7edf-11f0-8a36-06453a64f325\",
  \"EventId\": \"WebServerSecurityGroup-
CREATE_IN_PROGRESS-2025-08-21T22:37:22.626Z\",
  \"StackName\": \"my-stack\",
  \"LogicalResourceId\": \"WebServerSecurityGroup\",
  \"PhysicalResourceId\": \"my-stack-WebServerSecurityGroup-n8A43bQT1ty2\",
  \"ResourceType\": \"AWS::EC2::SecurityGroup\",
  \"Timestamp\": \"2025-08-21T22:37:22.626000+00:00\",
  \"ResourceStatus\": \"CREATE_IN_PROGRESS\",
  \"ResourceStatusReason\": \"Resource creation Initiated\",
  \"ResourceProperties\": \"{\\\"GroupDescription\\\": \\"Allow HTTP access from my IP
address\\\", \\"SecurityGroupIngress\\\": [{\\\"CidrIp\\\": \\"0.0.0.0/0\\\", \\"Description\\\": \\"HTTP
\\\", \\"FromPort\\\": \\"80\\\", \\"ToPort\\\": \\"80\\\", \\"IpProtocol\\\": \\"tcp\\\"}]]}\",
  \"ClientRequestToken\": \"token\"
},
{
  \"StackId\": \"arn:aws:cloudformation:aws-region:123456789012:stack/my-
stack/64726230-7edf-11f0-8a36-06453a64f325\",
  \"EventId\": \"WebServerSecurityGroup-
CREATE_IN_PROGRESS-2025-08-21T22:37:20.186Z\",
  \"StackName\": \"my-stack\",
  \"LogicalResourceId\": \"WebServerSecurityGroup\",

```

```
    "PhysicalResourceId": "",
    "ResourceType": "AWS::EC2::SecurityGroup",
    "Timestamp": "2025-08-21T22:37:20.186000+00:00",
    "ResourceStatus": "CREATE_IN_PROGRESS",
    "ResourceProperties": "{\"GroupDescription\":\"Allow HTTP access from my IP address\", \"SecurityGroupIngress\": [{\"CidrIp\":\"0.0.0.0/0\", \"Description\":\"HTTP\", \"FromPort\":\"80\", \"ToPort\":\"80\", \"IpProtocol\":\"tcp\"}]}",
    "ClientRequestToken": "token"
  },
  {
    "StackId": "arn:aws:cloudformation:aws-region:123456789012:stack/my-stack/64726230-7edf-11f0-8a36-06453a64f325",
    "EventId": "64740fe0-7edf-11f0-8a36-06453a64f325",
    "StackName": "my-stack",
    "LogicalResourceId": "my-stack",
    "PhysicalResourceId": "arn:aws:cloudformation:aws-region:123456789012:stack/my-stack/64726230-7edf-11f0-8a36-06453a64f325",
    "ResourceType": "AWS::CloudFormation::Stack",
    "Timestamp": "2025-08-21T22:37:17.819000+00:00",
    "ResourceStatus": "CREATE_IN_PROGRESS",
    "ResourceStatusReason": "User Initiated",
    "ClientRequestToken": "token"
  }
]
```

Los eventos más recientes se informan en primer lugar. La siguiente tabla describe los campos devueltos por el comando describe-stack-events:

Campo	Descripción
EventId	Identificador de eventos.
StackName	Nombre de la pila a la que corresponde el evento.
StackId	Identificador de la pila a la que corresponde el evento.
LogicalResourceId	Identificador lógico del recurso.
PhysicalResourceId	Identificador físico del recurso.
ResourceProperties	Propiedades del recurso.

Campo	Descripción
ResourceType	Tipo del recurso.
Timestamp	Hora en la que se produjo el evento.
ResourceStatus	El estado del recurso, que puede ser uno de los siguientes códigos de estado: 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 El estado DELETE_SKIPPED se aplica a los recursos con un atributo de la política de eliminación de conservación.
DetailedStatus	Estado detallado de la pila. Si CONFIGURATION_COMPLETE está presente, la fase de configuración de los recursos de la pila se ha completado y la estabilización de los recursos está en curso.
ResourceStatusReason	Más información sobre el estado.

Códigos de estado de pilas

En la siguiente tabla se describen los códigos de estado de pila:

Estado de la pila y estado detallado opcional	Descripción
CREATE_COMPLETE	Creación correcta de una o varias pilas.
CREATE_IN_PROGRESS	Creación continua de una o varias pilas.
CREATE_FAILED	Creación incorrecta de una o varias pilas. Consulte los eventos de pila para ver los mensajes de error asociados . Algunos de los posibles motivos de que la creación no se haya realizado correctamente incluyen la falta de permisos para trabajar con todos los recursos de pila, el rechazo de los valores de los parámetros por parte de un servicio de AWS o que se ha agotado el tiempo de espera durante la creación del recurso.
DELETE_COMPLETE	Eliminación correcta de una o varias pilas. Las pilas eliminadas se conservan y se pueden ver durante 90 días.
DELETE_FAILED	Eliminación incorrecta de una o varias pilas. Dado que la eliminación no se ha realizado correctamente, es posible que algunos recursos aún se sigan ejecutando; sin embargo, no puede trabajar ni actualizar la pila. Vuelva a eliminar la pila o consulte los eventos de la pila para ver los mensajes de error asociados.
DELETE_IN_PROGRESS	Eliminación continua de una o varias pilas.
REVIEW_IN_PROGRESS	Creación continua de una o varias pilas con un StackId esperado, pero sin plantillas ni recursos.  Important Una pila con este código de estado cuenta para el número máximo posible de pilas.
ROLLBACK_COMPLETE	Eliminación correcta de una o varias pilas después de un error en la creación de la pila o después de que se haya cancelado

Estado de la pila y estado detallado opcional	Descripción
	explícitamente la creación de la pila. La pila vuelve al estado anterior de trabajo. Se eliminan todos los recursos que se hayan creado durante la acción de creación de pila. Este estado solo existe después de la creación fallida de una pila. Esto significa que todas las operaciones de la pila creada parcialmente se ha limpiado de forma apropiada. Cuando en este estado, solo puede realizarse una operación de eliminación.
ROLLBACK_FAILED	Eliminación incorrecta de una o varias pilas después de un error en la creación de pila o después de que se haya cancelado explícitamente la creación de pila. Elimine la pila o consulte los eventos de pila para ver los mensajes de error asociados.
ROLLBACK_IN_PROGRESS	Eliminación continua de una o varias pilas después de un error en la creación de pila o después de que se haya cancelado explícitamente la creación de pila.
UPDATE_COMPLETE	Actualización correcta de una o varias pilas.
UPDATE_COMPLETE_CLEANUP_IN_PROGRESS	Eliminación continua de recursos antiguos para una o varias pilas después de una actualización correcta de las pilas. Para las actualizaciones de pilas que requieren que se sustituyan los recursos, CloudFormation crea los nuevos recursos en primer lugar y, a continuación, elimina los antiguos para ayudar a reducir interrupciones con la pila. En este estado, la pila se ha actualizado y se puede utilizar, pero CloudFormation sigue eliminando los recursos antiguos.
UPDATE_FAILED	Actualización incorrecta de una o varias pilas. Consulte los eventos de pila para ver los mensajes de error asociados.
UPDATE_IN_PROGRESS	Actualización continua de una o varias pilas.

Estado de la pila y estado detallado opcional	Descripción
UPDATE_ROLLBACK_COMPLETE	Recuperación correcta del estado activo anterior de una o varias pilas después de un error en la actualización de las pilas.
UPDATE_ROLLBACK_COMPLETE_CLEANUP_IN_PROGRESS	Eliminación continua de recursos nuevos para una o varias pilas después de una actualización incorrecta de las pilas. En este estado, se ha restaurado el estado anterior de la pila y se puede utilizar, pero CloudFormation sigue eliminando los nuevos recursos que ha creado durante la actualización de la pila.
UPDATE_ROLLBACK_FAILED	Restauración incorrecta del estado activo anterior de una o varias pilas después de un error en la actualización de las pilas. En este estado, puede eliminar la pila o continuar con la restauración . Probablemente necesite corregir errores antes de que la pila vuelva a estar operativa. También puede contactar con Soporte para restaurar la pila a un estado útil.
UPDATE_ROLLBACK_IN_PROGRESS	Recuperación continua del estado activo anterior de una o varias pilas después de un error en la actualización de las pilas.
IMPORT_IN_PROGRESS	La operación de importación está actualmente en curso.
IMPORT_COMPLETE	La operación de importación se ha completado correctamente para todos los recursos de la pila compatibles con <code>resource import</code> .
IMPORT_ROLLBACK_IN_PROGRESS	La importación volverá a la configuración de la plantilla anterior.
IMPORT_ROLLBACK_FAILED	La operación de restauración de importación ha generado un error para un recurso de la pila como mínimo. Los resultados estarán disponibles para los recursos que CloudFormation haya importado correctamente.

Estado de la pila y estado detallado opcional	Descripción
IMPORT_ROLLBACK_COMPLETE	La importación ha vuelto correctamente a la configuración de plantilla anterior.

Visualización de eventos de la pila por operación

Puede ver los eventos de la pila agrupados por operación para comprender mejor la secuencia y el alcance de los cambios realizados en la pila. La agrupación basada en operaciones le ayuda a realizar un seguimiento conjunto de los eventos relacionados, lo que facilita el seguimiento del progreso y la resolución de problemas cuando se producen.

A cada operación de la pila (creación, actualización, eliminación o reversión) se le asigna un ID de operación único que agrupa todos los eventos relacionados. Esto le permite centrarse en operaciones específicas e identificar rápidamente la causa raíz de los errores.

Temas

- [Requisitos previos](#)
- [Visualización de eventos de la pila por operación \(consola\)](#)
- [Visualización de eventos de la pila por operación \(AWS CLI\)](#)
- [Códigos de estado de pilas](#)

Requisitos previos

Para usar la API `DescribeEvents`, debe tener los permisos de IAM necesarios para: `DescribeEvents`.

Visualización de eventos de la pila por operación (consola)

Visualización de eventos de la pila agrupados por operación

1. Inicie sesión en la Consola de administración de AWS y abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation>.
2. En la barra de navegación de la parte superior de la pantalla, elija la Región de AWS en la que creó la pila.

3. En la página Pilas, seleccione el nombre de la pila. CloudFormation muestra la información de la pila que haya seleccionado.
4. Seleccione la pestaña Eventos para ver los eventos de la pila que CloudFormation generó para la pila.
5. Los eventos se agrupan automáticamente por ID de operación. Cada operación aparece como una sección ampliable que muestra el tipo, el estado y la marca temporal de la operación.
6. Haga clic en el ID de operación para abrir una vista detallada que muestre solo los eventos relacionados con esa operación específica.
7. En la vista de detalles de la operación, seleccione la casilla Mostrar solo errores para mostrar solo los eventos con errores en el análisis de la causa raíz.

CloudFormation actualiza automáticamente los eventos de la pila cada minuto. La insignia Nuevos eventos disponibles aparece cuando se producen nuevos eventos de la pila. Seleccione el icono de actualización para cargar estos eventos en la lista.

Al visualizar eventos de la pila agrupados por operación, puede comprender la secuencia de eventos de cada operación e identificar rápidamente qué operación específica causó problemas (si está depurando la pila).

Mientras la operación de la pila está en ejecución, aparece con el estado `CREATE_IN_PROGRESS`, `UPDATE_IN_PROGRESS` o `DELETE_IN_PROGRESS`. Una vez que la operación se haya completado correctamente, su estado cambiará a `CREATE_COMPLETE`, `UPDATE_COMPLETE` o `DELETE_COMPLETE`.

Para obtener más información, consulte [Comprensión de los eventos de creación de la pila de CloudFormation](#) y [Monitorización del progreso de una actualización de pila](#).

Visualización de eventos de la pila por operación (AWS CLI)

Puede usar el comando `describe-events` con el filtrado de ID de operación para ver los eventos de operaciones específicas.

Obtención de los ID de la última operación

La descripción de la pila, disponible a través de la API `describe-stacks`, ya incluye información de `LastOperations` que muestra los ID de operaciones recientes y sus tipos. Esto le permite identificar rápidamente qué operaciones se han realizado y su estado actual sin tener que analizar los registros de eventos.

```
aws cloudformation describe-stacks --stack-name MyStack
```

El siguiente es un ejemplo de respuesta que muestra que la última operación fue una reversión tras una operación de actualización fallida.

```
{
  "Stacks": [
    {
      "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/MyStack/07580010-bb79-11f0-8f6c-0289bb5c804f",
      "StackName": "MyStack",
      "Description": "A simple CloudFormation template to create an S3 bucket.",

      "CreationTime": "2025-11-07T01:28:13.778000+00:00",
      "LastUpdatedTime": "2025-11-07T01:43:39.838000+00:00",
      "RollbackConfiguration": {},
      "StackStatus": "UPDATE_ROLLBACK_COMPLETE",
      "DisableRollback": false,
      "NotificationARNs": [],
      "Tags": [],
      "EnableTerminationProtection": false,
      "DriftInformation": {
        "StackDriftStatus": "NOT_CHECKED"
      },
      "LastOperations": [
        {
          "OperationType": "ROLLBACK",
          "OperationId": "d0f12313-7bdb-414d-a879-828a99b36f29"
        },
        {
          "OperationType": "UPDATE_STACK",
          "OperationId": "1c211b5a-4538-4dc9-bfed-e07734371e57"
        }
      ]
    }
  ]
}
```

Filtrado de eventos por ID de operación

El siguiente comando `describe-events` describe los eventos de un ID de operación específico:

```
aws cloudformation describe-events \  
  --operation-id 1c211b5a-4538-4dc9-bfed-e07734371e57
```

Para ver solo los eventos fallidos para la solución de problemas, utilice el parámetro `--filter FailedEvents=true`:

```
aws cloudformation describe-events \  
  --operation-id 1c211b5a-4538-4dc9-bfed-e07734371e57 \  
  --filter FailedEvents=true
```

La nueva función de filtrado por ID de operación le permite centrarse en operaciones específicas y sus eventos relacionados. Resulta especialmente útil para lo siguiente:

- Solución de problemas específicos: aísle los eventos de una operación fallida para comprender qué es lo que ha fallado.
- Supervisión de las operaciones de larga duración: realice un seguimiento del progreso de las actualizaciones complejas o de las implementaciones de pilas de gran tamaño.
- Auditoría de cambios: revise todos los eventos asociados a una operación de actualización concreta.
- Análisis de la causa raíz: utilice el filtro de errores para identificar rápidamente el origen de los problemas de implementación.

Códigos de estado de pilas

En la siguiente tabla se describen los campos devueltos por el comando `describe-events` al utilizar el filtrado por ID de operación:

Campo	Descripción
EventId	Identificador de eventos.
OperationId	Identificador único de la operación que generó este evento.
StackName	Nombre de la pila a la que corresponde el evento.
StackId	Identificador de la pila a la que corresponde el evento.

Campo	Descripción
LogicalResourceId	Identificador lógico del recurso.
PhysicalResourceId	Identificador físico del recurso.
ResourceProperties	Propiedades del recurso.
ResourceType	Tipo del recurso.
Timestamp	Hora en la que se produjo el evento.
ResourceStatus	El estado del recurso (CREATE_COMPLETE , UPDATE_FAILED , etc.).
DetailedStatus	Estado detallado de la pila. Si CONFIGURATION_COMPLETE está presente, la fase de configuración de los recursos de la pila se ha completado y la estabilización de los recursos está en curso.
ResourceStatusReason	Más información sobre el estado.

Visualización de un cronograma de la implementación de una pila de CloudFormation

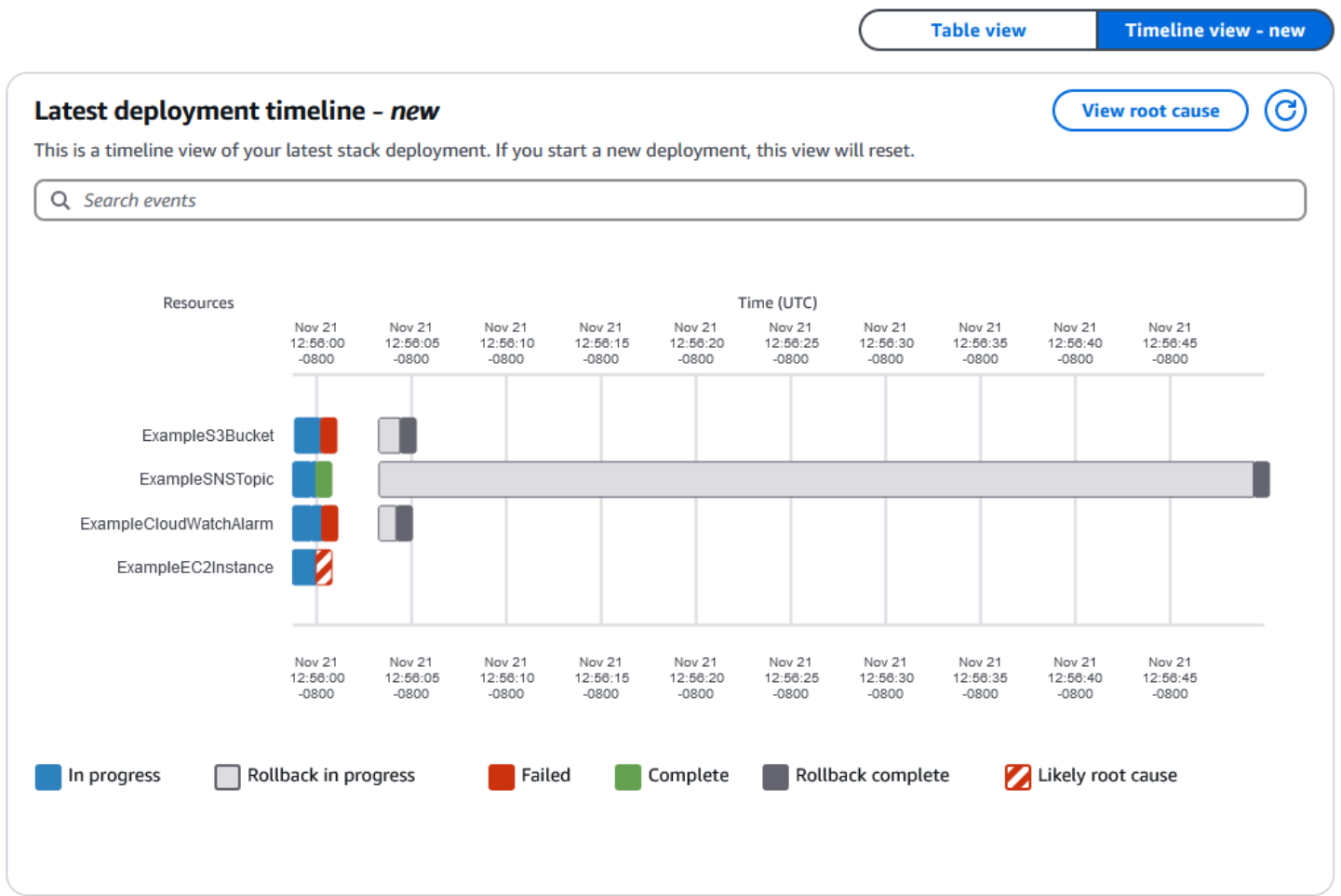
El gráfico del cronograma de implementación de la pila proporciona una representación visual de una línea de tiempo de implementación de la pila. Esta vista muestra los estados de implementación de la pila y de cada uno de sus recursos, así como las veces que ha cambiado cada estado. Los estados de implementación de la pila se representan con el color correspondiente.

Temas

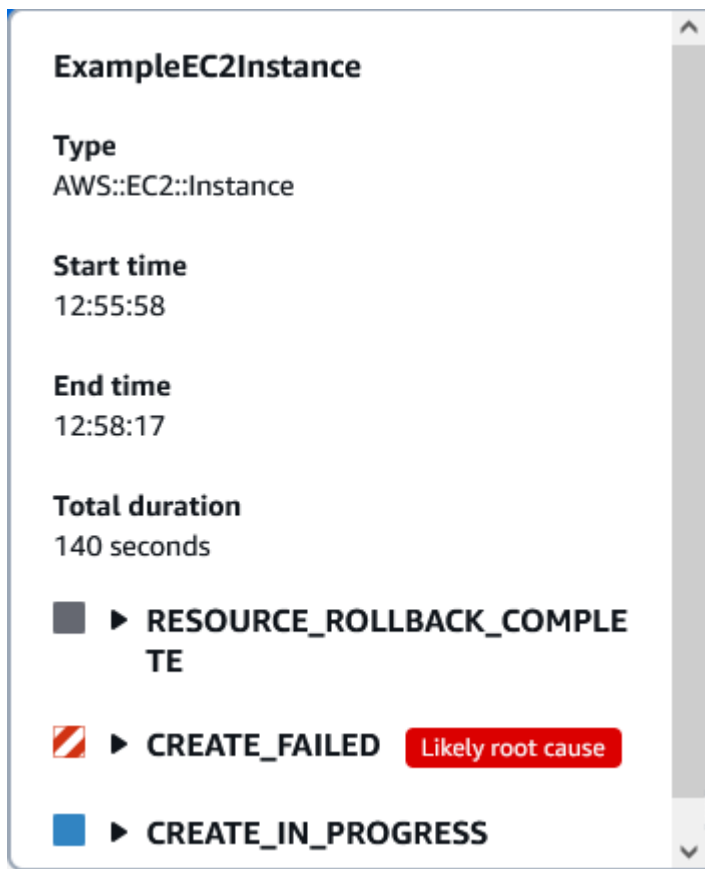
- [Comprensión del gráfico del cronograma de implementación de la pila](#)
- [Visualización del gráfico del cronograma de implementación de la pila \(consola\)](#)

Comprensión del gráfico del cronograma de implementación de la pila

La siguiente imagen muestra el gráfico del cronograma de una implementación de pila que falló debido a un error en el lanzamiento de un recurso de instancia de Amazon EC2.



La elección de cada recurso proporciona información más detallada sobre el cronograma de implementación:



Al seleccionar un recurso, se muestran el Tipo, la Hora de inicio de la implementación, la Hora de finalización de la implementación y la Duración total de la implementación. También encontrará la Hora de inicio, la Hora de finalización y la Duración de cada estado de implementación en los menús desplegados que aparecen a continuación. Si el recurso no se pudo implementar, se indicará el Motivo del error.

Para obtener más información sobre estados de pilas, consulte [Códigos de estado de pilas](#).

Visualización del gráfico del cronograma de implementación de la pila (consola)

Para ver un gráfico del cronograma de implementación de la pila:

1. Abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation>.
2. En la barra de navegación de la parte superior de la pantalla, elija la Región de AWS en la que creó la pila.
3. En la página Pilas de la consola de CloudFormation, seleccione el nombre de la pila. CloudFormation muestra la información de la pila que haya seleccionado.

4. Seleccione la pestaña Eventos para ver los eventos de la pila que CloudFormation generó para la pila.
5. Pulse el botón Gráfico del cronograma para ver el gráfico del cronograma de su pila.

Comprensión de los eventos de creación de la pila de CloudFormation

Durante la implementación de la pila, se producen varios eventos para crear, configurar y validar los recursos definidos en la plantilla de la pila. Comprender estos eventos puede ayudar con la optimización del proceso de creación de pilas y a agilizar las implementaciones.

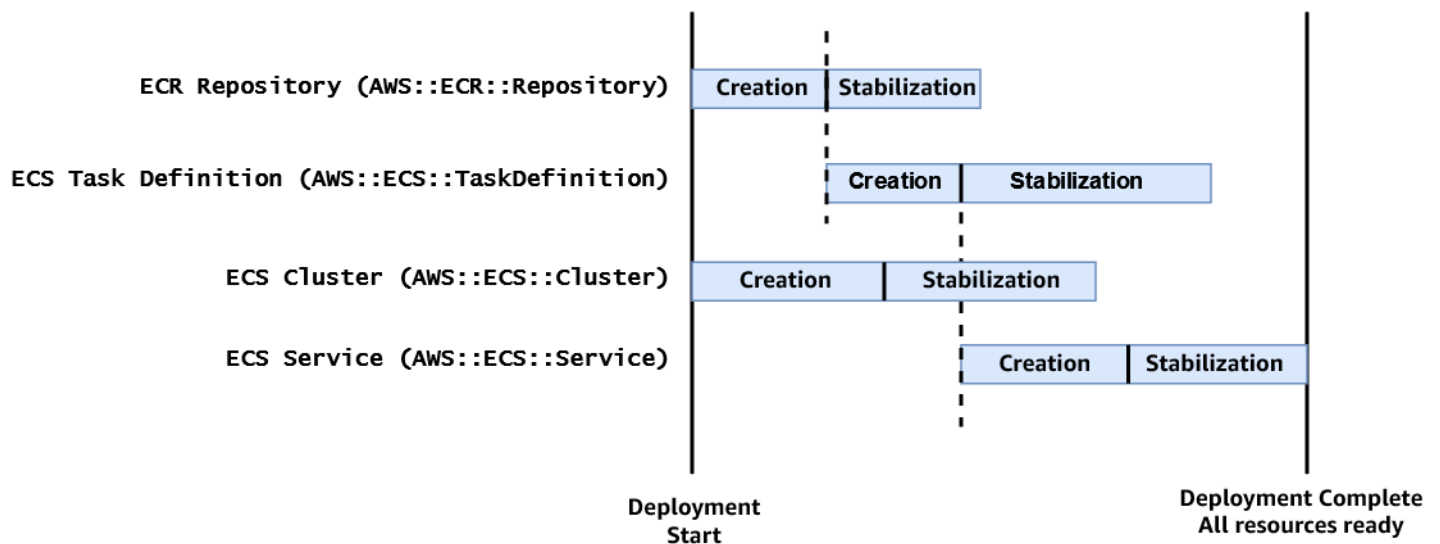
- **Eventos de creación de recursos:** cuando cada recurso inicia el proceso de creación, se establece el estado del `CREATE_IN_PROGRESS` evento. Este evento indica que el recurso se está aprovisionando.
- **Comprobación de coherencia eventual:** gran parte del tiempo de creación de la pila se dedica a realizar una eventual comprobación de coherencia con los recursos creados por la pila. Durante esta fase, el servicio realiza comprobaciones de coherencia internas para garantizar que el recurso esté en pleno funcionamiento y cumpla con los criterios de estabilización del servicio definidos por todos los Servicio de AWS.
- **Evento completo de configuración:** cuando cada recurso finaliza la fase final de verificación de coherencia del aprovisionamiento, se establece un estado detallado del evento `CONFIGURATION_COMPLETE`.
- **Evento completo de creación de recurso:** una vez que el recurso se haya creado y configurado según lo especificado y la configuración coincida con lo especificado en la plantilla, se establece el estado del evento `CREATE_COMPLETE`.

Puede aprovechar el evento `CONFIGURATION_COMPLETE` para agilizar el proceso de creación de pilas en situaciones en las que no sea necesario comprobar la coherencia de los recursos, como la validación de una configuración de pila previa a la producción o el aprovisionamiento entre pilas. Puede utilizar este evento de varias maneras. Por ejemplo, puedes usarlo como una señal visual para evitar esperar a que finalice la comprobación de coherencia de los recursos o de la pila. O puede usarlo para crear un mecanismo automatizado mediante integración y entrega continua (CI/CD) para activar acciones adicionales.

⚠ Important

Si bien aprovechar el evento `CONFIGURATION_COMPLETE` acelera los tiempos de creación de las pilas, debes tener en cuenta sus ventajas y desventajas. En primer lugar, solo es compatible con un subconjunto de tipos de recursos que admiten la detección de desviaciones. Para obtener una lista de tipos de recursos que admiten la detección de desviaciones, consulte [Tipo de recurso compatible](#). Es posible que este enfoque no sea adecuado para todos los escenarios, especialmente cuando los recursos requieren, en última instancia, exhaustivas comprobaciones de coherencia para garantizar la plena preparación operativa en todo el entorno de nube (por ejemplo, en los entornos de producción). Le recomendamos evaluar detenidamente sus requisitos de implementación y la importancia de las comprobaciones de coherencia de cada recurso. Aproveche el evento `CONFIGURATION_COMPLETE` para optimizar las velocidades de implementación sin comprometer la integridad y la fiabilidad de su infraestructura.

Como no se garantiza que el evento `CONFIGURATION_COMPLETE` esté configurado, cualquier escenario que lo utilice debe estar preparado para gestionar un evento `CREATE_COMPLETE` en el que no se haya establecido ningún evento `CONFIGURATION_COMPLETE`.



Cuando se inicia la implementación de la pila, tanto los recursos `AWS::ECR::Repository` como los recursos `AWS::ECS::Cluster` inician el proceso de creación (`ResourceStatus = CREATE_IN_PROGRESS`). Cuando el tipo `AWS::ECR::Repository` de recurso haya iniciado la eventual comprobación de coherencia (`DetailedStatus = CONFIGURATION_COMPLETE`),

entonces, el recurso `AWS::ECS::TaskDefinition` podrá iniciar el proceso de creación. Del mismo modo, una vez que el recurso `AWS::ECS::TaskDefinition` comience la eventual verificación de coherencia, el recurso `AWS::ECS::Service` iniciará el proceso de creación.

Eventos de **CREATE_IN_PROGRESS** y **CREATE_COMPLETE**

- [Pila]: **CREATE_IN_PROGRESS**
- [Recurso]: Repositorio ECR **CREATE_IN_PROGRESS**
- [Recurso]: Clúster ECS **CREATE_IN_PROGRESS**
- [Recurso]: Repositorio ECR **CREATE_IN_PROGRESS**, **CONFIGURATION_COMPLETE**
- [Recurso]: Definición de la tarea de ECS **CREATE_IN_PROGRESS**
- [Recurso]: Clúster ECS **CREATE_IN_PROGRESS**, **CONFIGURATION_COMPLETE**
- [Recurso]: Definición de la tarea de ECS **CREATE_IN_PROGRESS**, **CONFIGURATION_COMPLETE**
- [Recurso]: Servicio ECS **CREATE_IN_PROGRESS**
- [Recurso]: Repositorio ECR **CREATE_COMPLETE**
- [Recurso]: Clúster ECS **CREATE_COMPLETE**
- [Recurso]: Servicio ECS **CREATE_IN_PROGRESS**, **CONFIGURATION_COMPLETE**
- [Pila]: **CREATE_IN_PROGRESS**, **CONFIGURATION_COMPLETE**
- [Recurso]: Definición de la tarea de ECS **CREATE_COMPLETE**
- [Recurso]: Servicio ECS **CREATE_COMPLETE**
- [Pila]: **CREATE_COMPLETE**

Monitorización del progreso de una actualización de pila

Puede monitorizar el progreso de una actualización de pila visualizando los eventos de la pila. La pestaña Eventos de la consola muestra todos los pasos importantes de la creación y actualización de la pila ordenados por hora de cada evento, con los últimos eventos en la parte superior. Para obtener más información, consulte [Monitorizar el progreso de la pila](#).

Temas

- [Eventos generados durante una actualización de pila correcta](#)
- [Eventos generados cuando falla la actualización de un recurso](#)

Eventos generados durante una actualización de pila correcta

El inicio del proceso de actualización de la pila se marca con un evento `UPDATE_IN_PROGRESS` para la pila:

```
2011-09-30 09:35 PDT AWS::CloudFormation::Stack MyStack UPDATE_IN_PROGRESS
```

Los siguientes son eventos que marcan el comienzo y la realización de la actualización de cada recurso que se cambió en la plantilla de actualización. Por ejemplo, al actualizar un recurso [AWS::RDS::DBInstance](#) denominado MyDB, se producirían las siguientes entradas:

```
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
```

El evento `UPDATE_IN_PROGRESS` se registra cuando CloudFormation informa que ha comenzado a actualizar el recurso. El evento `UPDATE_COMPLETE` se registra cuando se crea correctamente el recurso.

Cuando CloudFormation haya actualizado correctamente la pila, verá el siguiente evento:

```
2011-09-30 09:35 PDT AWS::CloudFormation::Stack MyStack UPDATE_COMPLETE
```

Important

Durante las operaciones de actualización de la pila, si CloudFormation necesita reemplazar un recurso existente, primero crea un nuevo recurso y, a continuación, elimina el recurso antiguo. Sin embargo, puede haber casos en los que CloudFormation no puede eliminar el recurso antiguo (por ejemplo, si el usuario no tiene permisos para eliminar un recurso de un tipo determinado).

CloudFormation realiza tres intentos para eliminar el recurso antiguo. Si CloudFormation no puede eliminar el recurso antiguo, lo elimina de la pila y continúa actualizando la pila. Cuando se completa la actualización de la pila, CloudFormation emite un evento de pila `UPDATE_COMPLETE`, pero incluye un `StatusReason` que indica que uno o más recursos no se pudieron eliminar. CloudFormation también emite un evento `DELETE_FAILED` para el recurso específico, con un `StatusReason` correspondiente que proporciona más detalles sobre por qué CloudFormation no pudo eliminar el recurso.

El recurso antiguo aún existe y seguirá incurriendo en cargos, pero ya no es accesible a través de CloudFormation. Para eliminar el recurso antiguo, acceda al recurso antiguo directamente mediante la consola o la API del servicio subyacente.

Esto también es cierto para los recursos que ha eliminado de la plantilla de pila, por lo que se eliminará de la pila durante la actualización de la pila.

Eventos generados cuando falla la actualización de un recurso

Si la actualización de un recurso falla, CloudFormation informa de un evento `UPDATE_FAILED` que incluye un motivo del error. Por ejemplo, si su plantilla de actualización especificó un cambio de propiedad que no es compatible con el recurso, como la reducción del tamaño de `AllocatedStorage` para un recurso [AWS::RDS::DBInstance](#), se verían eventos como estos:

```
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
```

Si falla la actualización de un recurso, CloudFormation revierte cualquier recurso actualizado durante la actualización a sus configuraciones previas. A continuación se muestra un ejemplo de los eventos que se verían durante una restauración de actualización:

```
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]
```

Continuación de la restauración de una actualización

A veces, cuando CloudFormation intenta revertir una actualización de pila, no puede revertir todos los cambios que realizó durante el proceso de actualización. Esto se denomina estado `UPDATE_ROLLBACK_FAILED`. Por ejemplo, puede tener una pila que comience a restaurar una antigua instancia de base de datos que se eliminó fuera de CloudFormation. Dado que CloudFormation no sabe que la base de datos se ha eliminado, asume que la instancia de base de datos aún existe e intenta restaurarla, lo que provoca que la restauración de la actualización falle.

Una pila en el estado `UPDATE_ROLLBACK_FAILED` no se puede actualizar, pero se puede revertir a un estado de funcionamiento (`UPDATE_ROLLBACK_COMPLETE`). Luego de devolver la pila a su configuración original, puede tratar de volver a actualizarla.

En la mayoría de los casos, debe solucionar el error que hace que la restauración de la actualización genere un error antes de continuar la restauración de la pila. En otros casos, puede continuar la restauración de la actualización sin ningún cambio, por ejemplo cuando se agota el tiempo de espera de una operación de pila.

 Note

Si utiliza pilas anidadas, al restaurar la pila principal se intentará restaurar también todas las pilas secundarias.

Para continuar la restauración de una actualización (consola)

1. Inicie sesión en la Consola de administración de AWS y abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation>.
2. En la barra de navegación de la parte superior de la pantalla, elija la Región de AWS en donde se encuentra la pila.
3. En la página Pilas, seleccione la pila que desea actualizar, elija Acciones de pila y, a continuación, elija Continuar la restauración de la actualización.

Si no funciona ninguna de las soluciones de [Solución de errores](#), puede utilizar la opción avanzada para omitir los recursos que CloudFormation no puede restaurar correctamente. Debe [buscar](#) y escribir los ID lógicos de los recursos que desea omitir. Especifique únicamente los recursos que han pasado al estado `UPDATE_FAILED` durante la operación `UpdateRollback` y no durante la actualización de reenvío.

 Warning

CloudFormation establece el estado de los recursos especificados como `UPDATE_COMPLETE` y continúa restaurando la pila. Después de que la restauración se haya completado, el estado de los recursos omitidos será incompatible con el estado de los recursos de la plantilla de pila. Antes de realizar otra actualización de pila, debe

actualizar la pila o los recursos para que sean coherentes entre sí. De lo contrario, las actualizaciones de pilas posteriores podrían fallar y la pila será irrecuperable.

Especifique el número mínimo de recursos necesarios para restaurar correctamente la pila. Por ejemplo, una actualización de recursos incorrecta podría provocar que los recursos dependientes fallaran. En este caso, no sería necesario omitir los recursos dependientes.

Para omitir los recursos que forman parte de pilas anidadas, utilice el siguiente formato: *NestedStackName.ResourceLogicalID*. Si desea especificar el ID lógico de un recurso de pila (Type: `AWS::CloudFormation::Stack`) en la lista `ResourcesToSkip`, su pila integrada correspondiente debe estar en uno de los siguientes estados: `DELETE_IN_PROGRESS`, `DELETE_COMPLETE` o `DELETE_FAILED`.

Para continuar con la restauración de una actualización (AWS CLI)

- Utilice el comando [continue-update-rollback](#) con la opción `--stack-name` para especificar el ID de la pila que desea seguir restaurando.

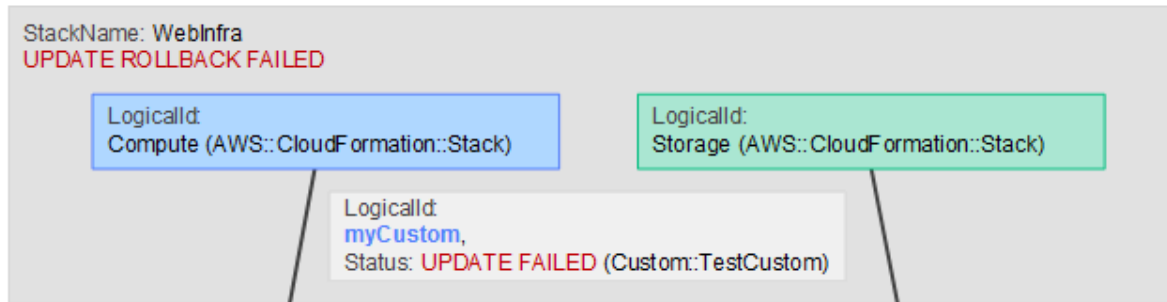
Continúe con la reversión de las actualizaciones con fallas de la pila anidada

Cuando tiene varias pilas anidadas dentro de cada una, es posible que tenga que omitir los recursos entre varios niveles anidados para que toda la jerarquía de pilas vuelva a funcionar correctamente.

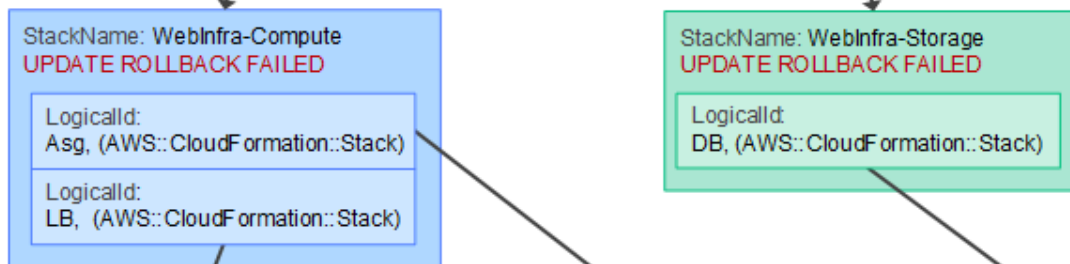
Por ejemplo, tiene una pila raíz llamada `WebInfra` que contiene dos pilas más pequeñas en su interior: `WebInfra-Compute` y `WebInfra-Storage`. Estas dos pilas también tienen sus propias pilas anidadas dentro de ellas.

Si algo sale mal durante una actualización y el proceso de actualización falla, es posible que toda la jerarquía de pilas acabe con el estado `UPDATE_ROLLBACK_FAILED`, como se muestra en el siguiente diagrama.

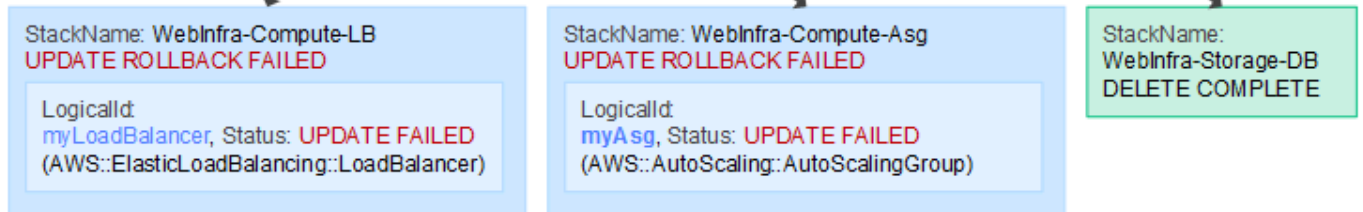
Root



Level 1



Level 2

**Note**

Los nombres de pilas de este ejemplo se han cortado para simplificar. CloudFormation suele generar nombres de pila secundarios, que contienen cadenas aleatorias únicas, por lo que es posible que los nombres reales no sean intuitivos.

Para poner la pila raíz en un estado de funcionamiento con el comando `continue-update-rollback`, debe utilizar la opción `--resources-to-skip` para omitir recursos que no se pudieron restaurar.

El ejemplo de `continue-update-rollback` a continuación reanuda una operación de reversión a partir de un intento fallido previo de una actualización de pila. En este ejemplo, la opción `--resources-to-skip` incluye los siguientes elementos:

- *myCustom*
- *WebInfra-Compute-Asg.myAsg*
- *WebInfra-Compute-LB.myLoadBalancer*
- *WebInfra-Storage.DB*

Para los recursos de la pila raíz, solo debe proporcionar el ID lógico, por ejemplo, *myCustom*. Sin embargo, para los recursos que se encuentran en las pilas anidadas, debe proporcionar el nombre de la pila anidada y su ID lógico, separados por un punto. Por ejemplo, *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
```

Búsqueda del nombre de pila de una pila anidada

Puede ubicarlo dentro del ID de pila o Nombre de recurso de Amazon (ARN) de una pila secundaria.

El siguiente ejemplo de ARN refiere a una pila denominada `WebInfra-Storage-Z2VKC706XKXT`.

```
arn:aws:cloudformation:us-east-1:123456789012:stack/WebInfra-Storage-Z2VKC706XKXT/  
ea9e7f90-54f7-11e6-a032-028f3d2330bd
```

Búsqueda del ID lógico de una pila anidada

Puede encontrar el ID lógico de una pila secundaria en la definición de plantilla de su principal. En el diagrama, el `LogicalId` de la pila secundaria `WebInfra-Storage-DB` es `DB` en su principal `WebInfra-Storage`.

En la consola de CloudFormation, también puede encontrar el ID lógico en la columna Logical ID (ID lógico) del recurso de pila en la pestaña Resources (Recursos) o la pestaña Events (Eventos). Para obtener más información, consulte [Visualización de la información de la pila desde la consola de CloudFormation](#).

Determine la causa de una falla en la pila

Si no se puede crear la pila, CloudFormation puede ser de ayuda para determinar el evento que probablemente sea la causa raíz del error de la pila. Según el escenario y sus permisos, es posible que los eventos AWS CloudTrail puedan proporcionar más detalles sobre la causa raíz en caso de que el motivo del estado proporcionado en Eventos no sea claro.

Determinación de la causa principal de una falla en la pila

1. Abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation>.
2. En la página Pilas, seleccione la pila con una falla.
3. Seleccione la pestaña Eventos.
4. Seleccione Detectar causa raíz. CloudFormation analizará el error e indicará el evento que probablemente sea la causa raíz del error al agregar la etiqueta Causa raíz probable al evento específico Estado. Consulte Motivo del estado para obtener una explicación más detallada del estado en la consola de CloudFormation.
5. Elija el estado con errores con la etiqueta Causa raíz probable para obtener más información sobre la causa del error. Según el escenario y sus permisos, es posible que pueda revisar un evento de CloudTrail detallado. Estos son los siguientes posibles resultados de la elección del estado:
 - Los eventos de CloudTrail relacionados con este problema están disponibles y pueden ayudar a resolverlo. Visualice los eventos de CloudTrail.
 - No hemos encontrado ningún evento de CloudTrail relacionado con este problema que pueda ayudar a resolverlo.
 - Sus permisos actuales no permiten el acceso para ver los eventos de CloudTrail. Más información.
 - En proceso de comprobar los eventos de CloudTrail disponibles, vuelva a comprobarlos en unos minutos.
 - Se produjo un error al buscar los eventos de CloudTrail. Para una inspección manual, visite la consola de CloudTrail.
6. Si el motivo indicado en Motivo del estado no está claro y la causa raíz muestra un enlace a la consola de CloudTrail, abra el enlace para ver el evento y encontrar una causa raíz detallada.

Para obtener más información sobre los eventos de CloudTrail, consulte [Understanding CloudTrail events](#) y [CloudTrail record contents](#).

Para obtener más información sobre el historial de eventos de CloudTrail, consulte [Working with CloudTrail Event history](#).

 Note

Las pilas anidadas no admiten la opción Detectar causa raíz.

Elija cómo gestionar los errores al aprovisionar recursos

Si se produce un error en la operación de la pila, no tendrá que restaurar los recursos que ya se habían aprovisionado correctamente y volver a empezar desde el principio cada vez. En su lugar, puede solucionar los problemas de los recursos en un estado `UPDATE_FAILED` o `CREATE_FAILED` y, a continuación, reanudar el aprovisionamiento desde el punto en el que se produjo el problema.

Para ello, debe activar la opción para conservar los recursos aprovisionados correctamente. Esta opción está disponible para todas las implementaciones de pilas y operaciones de conjuntos de cambios.

- Para crear pilas, si elige la opción Conservar los recursos aprovisionados correctamente, CloudFormation conserva el estado de los recursos que se crearon correctamente y deja los con errores en un estado con error hasta que se lleve a cabo la siguiente operación de actualización.
- Actualizar y cambiar las operaciones de conjunto establecidas en Conservar los recursos aprovisionados correctamente mantiene el estado de los recursos satisfactorios y, al mismo tiempo, reduce los recursos con errores al último estado estable conocido. El estado de los recursos con error será `UPDATE_FAILED`. Los recursos sin un último estado estable conocido se eliminarán en la siguiente operación de pila.

Temas

- [Descripción general de las opciones de errores de pila](#)
- [Condiciones necesarias para pausar la reversión de la pila](#)
- [Conservar los recursos aprovisionados correctamente \(consola\)](#)
- [Conservar los recursos aprovisionados correctamente \(AWS CLI\)](#)

Descripción general de las opciones de errores de pila

Antes de emitir una operación desde la consola de CloudFormation, la API o la AWS CLI, especifique el comportamiento de los errores de recursos aprovisionados. A continuación, continúe con el proceso de implementación de sus recursos sin ninguna otra modificación. En caso de que se produzca un error operativo, CloudFormation se detiene en el primer error de cada ruta de aprovisionamiento independiente. CloudFormation identifica las dependencias entre los recursos para paralelizar acciones de aprovisionamiento independientes. A continuación, procede a aprovisionar recursos en cada ruta de aprovisionamiento independiente hasta que se produce un error. Un error en una ruta no afecta a otras rutas de aprovisionamiento. CloudFormation continuará aprovisionando los recursos hasta que se completen o se detengan en un error diferente.

Solucione cualquier problema para continuar con el proceso de implementación. CloudFormation lleva a cabo las actualizaciones necesarias antes de volver a intentar las acciones de aprovisionamiento en los recursos que no se pudieron aprovisionar correctamente antes. Para solucionar los problemas, envíe una operación Retry (Reintentar), Update (Actualizar) o Roll back (Restaurar). Por ejemplo, si va a aprovisionar una instancia de Amazon EC2 y se produce un error en la instancia de EC2 durante una operación de creación, es posible que desee investigar el error, en lugar de deshacer el recurso fallido de inmediato. Puede revisar las verificaciones de estado del sistema y las verificaciones de estado de las instancias y, a continuación, seleccionar la operación Retry (Reintentar) una vez que se resuelvan los problemas.

Cuando se produce un error en una operación de pila y se ha especificado Preserve successfully provisioned resources (Conservar los recursos aprovisionados correctamente) en el menú Stack failure options (Opciones de error de pila), puede seleccionar las siguientes opciones.

- **Retry (Reintentar):** reintenta la operación de aprovisionamiento en los recursos con errores y continúa aprovisionando la plantilla hasta que se complete correctamente la operación de pila o se produzca el siguiente error. Seleccione esta opción si el recurso no se ha podido aprovisionar debido a un problema que no requiere modificaciones en la plantilla, como un permiso de AWS Identity and Access Management (IAM).
- **Update (Actualizar):** los recursos que se han aprovisionado se actualizan en las actualizaciones de la plantilla. Los recursos que no se hayan podido crear o actualizar se volverán a intentar. Seleccione esta opción si el recurso no se ha podido aprovisionar debido a errores de plantilla y ha modificado la plantilla. Cuando actualiza una pila cuyo estado es FAILED, debe seleccionar Preserve successfully provisioned resources (Conservar los recursos aprovisionados correctamente) en Stack failure options (Opciones de error de pila) para seguir actualizando la pila.
- **Roll back (Restaurar):** CloudFormation restaura la pila al último estado estable conocido.

Condiciones necesarias para pausar la reversión de la pila

Para evitar que CloudFormation revertiera y elimine de forma automática los recursos que se crearon correctamente, deben cumplirse las siguientes condiciones.

1. Al crear o actualizar la pila, debe elegir la opción Conservar los recursos aprovisionados correctamente. Esto le indica a CloudFormation que no elimine los recursos que se crearon correctamente, incluso si se produce un error en la operación general de la pila.
2. La operación de apilamiento debe haber fallado, lo que significa que el estado de la pila es `CREATE_FAILED` o `UPDATE_FAILED`.

Note

No se admiten los tipos de actualización inmutables.

Conservar los recursos aprovisionados correctamente (consola)

Create stack

Para conservar los recursos aprovisionados correctamente durante una operación de creación de pila

1. Inicie sesión en la Consola de administración de AWS y abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation>.
2. En la página Pilas, seleccione Crear pila en la parte superior derecha y, a continuación, seleccione Con recursos nuevos (estándar).
3. En Requisito previo: preparar la plantilla, elija Seleccione una plantilla existente.
4. En Especificar plantilla, elija especificar la dirección URL del bucket de S3 que contiene la plantilla de pila o cargar un archivo de plantilla de pila. A continuación, elija Siguiente.
5. En la página Specify stack details (Especificar los detalles de la pila), ingrese un nombre de pila en el cuadro Stack name (Nombre de la pila).
6. En la sección Parameters (Parámetros), especifique los parámetros que se definen en su plantilla de la pila.

Puede utilizar o cambiar cualquier parámetro con valores predeterminados.

7. Cuando esté satisfecho con los valores de los parámetros, elija Next (Siguiente).

8. En la página Configure stack options (Configurar opciones de pila), puede configurar opciones adicionales para su pila.
9. En Stack failure options (Opciones de error de pila), seleccione Preserve successfully provisioned resources (Conservar los recursos aprovisionados correctamente).
10. Cuando esté satisfecho con las opciones de pila, elija Next (Siguiente).
11. Revise su pila en la página Review (Revisar) y seleccione Create stack (Crear pila).

Resultados: los recursos que no se pudieron crear hacen la transición del estado de la pila a `CREATE_FAILED` para evitar que la pila se restaure cuando se produce un error en la operación de pila. El estado de los recursos que se aprovisionan correctamente es `CREATE_COMPLETE`. Puede supervisar la pila en la pestaña Stack events (Eventos de la pila).

Update stack

Para conservar los recursos aprovisionados correctamente durante una operación de pila de actualizaciones

1. Inicie sesión en la Consola de administración de AWS y abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation>.
2. Seleccione la pila que desea actualizar y, a continuación, Update (Actualizar).
3. En la página Update stack (Actualizar pila), elija una plantilla de pila utilizando una de las siguientes opciones:
 - Usar plantilla existente
 - Replace current template (Reemplazar la plantilla actual)
 - Editar la plantilla en Infrastructure Composer

Acepte la configuración y seleccione Next (Siguiente).

4. En la página Specify stack details (Especificar los detalles de la pila), especifique los parámetros que se definen en la plantilla de pila.

Puede utilizar o cambiar cualquier parámetro con valores predeterminados.

5. Cuando esté satisfecho con los valores de los parámetros, elija Next (Siguiente).
6. En la página Configure stack options (Configurar opciones de pila), puede configurar opciones adicionales para su pila.

7. Para el registro Behavior on provisioning failure (Comportamiento en caso de error de aprovisionamiento), seleccione Preserve successfully provisioned resources (Conservar los recursos aprovisionados correctamente).
8. Cuando esté satisfecho con las opciones de pila, elija Next (Siguiente).
9. Revise su pila en la página Review (Revisar) y seleccione Update stack (Actualizar pila).

Resultados: los recursos que no se pudieron actualizar hacen la transición del estado de la pila a UPDATE_FAILED y se restauran al último estado estable conocido. CloudFormation eliminará los recursos sin un último estado estable conocido en la siguiente operación de pila. El estado de los recursos que se aprovisionan correctamente es CREATE_COMPLETE o UPDATE_COMPLETE. Puede supervisar la pila en la pestaña Stack events (Eventos de la pila).

Change set

Note

Puede iniciar un conjunto de cambios para una pila con un estado de CREATE_FAILED o UPDATE_FAILED, pero no con un estado de UPDATE_ROLLBACK_FAILED.

Para conservar los recursos aprovisionados correctamente durante una operación de conjunto de cambios

1. Inicie sesión en la Consola de administración de AWS y abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation>.
2. Seleccione la pila que contenga el conjunto de cambios que desea iniciar y, a continuación, elija la pestaña Change sets (Conjuntos de cambios).
3. Seleccione el conjunto de cambios y, a continuación, elija Execute (Ejecutar).
4. Para Execute change set (Ejecutar conjunto de cambios), seleccione la opción Preserve successfully provisioned resources (Conservar los recursos aprovisionados correctamente).
5. Seleccione Execute change set (Ejecutar conjunto de cambios).

Resultados: los recursos que no se pudieron actualizar hacen la transición del estado de la pila a UPDATE_FAILED y se restauran al último estado estable conocido. CloudFormation eliminará los recursos sin un último estado estable conocido en la siguiente operación de pila. El estado de

los recursos que se aprovisionan correctamente es `CREATE_COMPLETE` o `UPDATE_COMPLETE`. Puede supervisar la pila en la pestaña Stack events (Eventos de la pila).

Conservar los recursos aprovisionados correctamente (AWS CLI)

Create stack

Para conservar los recursos aprovisionados correctamente durante una operación de creación de pila

Especifique la opción `--disable-rollback` o enumeración `on-failure DO_NOTHING` durante una operación [create-stack](#).

1. Proporcione un nombre de pila y una plantilla al comando `create-stack` con la opción `--disable-rollback`.

```
aws cloudformation create-stack --stack-name myteststack \  
  --template-body file://template.yaml \  
  --disable-rollback
```

El comando devuelve el siguiente resultado:

```
{  
  "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/  
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896"  
}
```

2. Describa el estado de la pila mediante el comando `describe-stacks`.

```
aws cloudformation describe-stacks --stack-name myteststack
```

El comando devuelve el resultado siguiente.

```
{  
  "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

Cómo preservar los recursos aprovisionados con éxito durante una operación de actualización de pila

1. Proporcione un nombre de pila y una plantilla existentes al comando `update-stack` con la opción `--disable-rollback`.

```

aws cloudformation update-stack --stack-name myteststack \
    --template-url https://s3.amazonaws.com/amzn-s3-demo-bucket/updated.template
    --disable-rollback

```

El comando devuelve el resultado siguiente.

```

{
    "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896"
}

```

2. Describa el estado de la pila mediante el comando `describe-stacks` o `describe-stack-events`.

```

aws cloudformation describe-stacks --stack-name myteststack

```

El comando devuelve el resultado siguiente.

```

{
    "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

Puede iniciar un conjunto de cambios para una pila con un estado de `CREATE_FAILED` o `UPDATE_FAILED`, pero no para un estado de `UPDATE_ROLLBACK_FAILED`.

Para conservar los recursos aprovisionados correctamente durante una operación de conjunto de cambios

Especifique la opción `--disable-rollback` durante una operación [execute-change-set](#).

1. Proporcione un nombre de pila y una plantilla al comando `execute-change-set` con la opción `--disable-rollback`.

```

aws cloudformation execute-change-set --stack-name myteststack \
  --change-set-name my-change-set --template-body file://template.yaml

```

El comando devuelve el siguiente resultado:

```

{
  "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. Inicie el conjunto de cambios con la opción `--disable-rollback`.

```
aws cloudformation execute-change-set --stack-name myteststack \  
  --change-set-name my-change-set --disable-rollback
```

3. Determine el estado de la pila mediante el comando `describe-stacks` o `describe-stack-events`.

```
aws cloudformation describe-stack-events --stack-name myteststack
```

El comando devuelve el resultado siguiente.

```
{  
  "StackEvents": [  
    {  
      "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/  
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896",  
      "EventId": "49c966a0-7b74-11ea-8071-024244bb0672",  
      "StackName": "myteststack",  
      "LogicalResourceId": " MyBucket",  
      "PhysicalResourceId": "myteststack-MyBucket-abcdefghijkl",  
      "ResourceType": "AWS::S3::Bucket",  
      "Timestamp": "2020-04-10T21:43:17.015Z",  
      "ResourceStatus": "UPDATE_FAILED"  
      "ResourceStatusReason": "User XYZ is not allowed to perform  
S3::UpdateBucket on MyBucket"  
    }  
  ]  
}
```

4. Corrija los errores de los permisos y vuelva a ejecutar la operación.

```
aws cloudformation update-stack --stack-name myteststack \  
  --use-previous-template --disable-rollback
```

El comando devuelve el resultado siguiente.

```
{  
  "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/  
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896"  
}
```

5. Describa el estado de la pila mediante el comando `describe-stacks` o `describe-stack-events`.

```
aws cloudformation describe-stacks --stack-name myteststack
```

El comando devuelve el resultado siguiente.

```
{
  "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
    }
  ]
}
```

Restauración de una pila

Puede usar el comando [rollback-stack](#) para restaurar una pila con un estado de pila `UPDATE_FAILED` o `CREATE_FAILED` a su último estado estable.

El siguiente comando `rollback-stack` revierte la pila especificada.

```
aws cloudformation rollback-stack --stack-name myteststack
```

El comando devuelve el resultado siguiente.

```
{
  "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896"
}
```

 Note

La operación `rollback-stack` borrará una pila si no contiene un último estado estable conocido.

Revierta su pila de CloudFormation en caso de violación de alarmas con activadores de reversión

Con los desencadenadores de restauración, puede hacer que CloudFormation supervise el estado de su aplicación durante la creación y la actualización de la pila y para restaurar dicha operación si la aplicación sobrepasa el umbral de las alarmas especificadas. Para cada desencadenador de restauración que crea, especifica la alarma de CloudWatch que CloudFormation debe supervisar. CloudFormation supervisa las alarmas especificadas durante la operación de creación o actualización de la pila y durante la cantidad de tiempo especificada una vez que se han implementado todos los recursos. Si alguna de las alarmas pasa al estado `ALARM` durante la operación de la pila o el periodo de monitoreo, CloudFormation revierte toda la operación de la pila.

Puede establecer un tiempo de monitorización desde el valor predeterminado de 0 hasta 180 minutos. Durante este tiempo, CloudFormation monitorea todos los desencadenadores de restauración una vez que la operación de creación o actualización de la pila implementa todos los recursos necesarios. Si alguna de las alarmas pasa al estado `ALARM` durante la operación de la pila o este periodo de monitoreo, CloudFormation revierte toda la operación de la pila. A continuación, para las operaciones de actualización, si el periodo de monitoreo termina sin que ninguna alarma pase al estado `ALARM`, CloudFormation procede a deshacerse de los recursos antiguos de la manera habitual. Si configura un tiempo de monitoreo pero no especifica desencadenadores de restauración, CloudFormation espera igualmente el periodo de tiempo especificado antes de limpiar los recursos antiguos para las operaciones de actualización. Puede utilizar este periodo de monitorización para llevar a cabo las validaciones manuales de la pila que desee y cancelar manualmente la creación o actualización de la pila según sea necesario. Si configura un tiempo de monitoreo de 0 minutos, CloudFormation monitorea igualmente los desencadenadores de restauración durante las operaciones de creación y actualización de la pila y revierte la operación si una alarma pasa al estado `ALARM`. A continuación, para las operaciones de actualización sin alarmas de incumplimiento, empieza a deshacerse de todos los recursos antiguos en cuanto se completa la operación.

De forma predeterminada, CloudFormation solo revierte las operaciones de pila si una alarma pasa al estado `ALARM`, no al `INSUFFICIENT_DATA`. Para que CloudFormation también revierta

la operación de la pila si una alarma pasa al estado `INSUFFICIENT_DATA`, edite la alarma de CloudWatch para tratar los datos que faltan como `breaching`. Para obtener más información, consulte [Configuración de cómo las alarmas de CloudWatch tratan los datos faltantes](#) en la Guía del usuario de Amazon CloudWatch.

CloudFormation no monitorea los desencadenadores de restauración cuando se revierte una pila durante una operación de actualización.

Puede agregar un máximo de 5 desencadenadores de restauración. Para agregar un desencadenador de restauración, especifique el nombre de recurso de Amazon (ARN) de la alarma de CloudWatch. Actualmente, los tipos `AWS::CloudWatch::Alarm` y `AWS::CloudWatch::CompositeAlarm` se pueden usar como desencadenadores de restauración. Para obtener más información acerca de las alarmas de CloudWatch, consulte [Uso de las alarmas de CloudWatch](#) en la Guía del usuario de Amazon CloudWatch.

Si falta una determinada alarma de CloudWatch, se produce un error en toda la operación de la pila y se restaura.

Tenga en cuenta que el acceso a CloudWatch requiere credenciales. Estas credenciales deben tener permisos para obtener acceso a los recursos de AWS, como la recuperación de datos de métricas de CloudWatch acerca de sus recursos. Para obtener más información, consulte [Autenticación y control de acceso para CloudWatch](#) en la Guía del usuario de Amazon CloudWatch.

Agregar disparadores de restauración durante la creación o actualización de la pila

Cómo agregar desencadenadores de restauración durante la creación o actualización de una pila (consola)

1. Durante la creación o actualización de una pila, en la página Configurar opciones de pila, en Opciones avanzadas, amplíe la sección Configuración de restauración.
2. Ingrese un tiempo de supervisión comprendido entre 0 y 180 minutos. El valor predeterminado es 0.
3. Especifique el ARN de la alarma o la alarma compuesta de CloudWatch que desee utilizar como desencadenador de restauración y seleccione Agregar el ARN de alarma de CloudWatch.

Por ejemplo, el siguiente es un ARN para una alarma de CloudWatch o una alarma compuesta, `arn:aws:cloudwatch:us-east-1:123456789012:alarm:MyAlarmName`.

4. Seleccione Siguiente y revise los detalles de la pila.
5. Cuando tenga todo listo, seleccione Enviar para crear o actualizar la pila.

Cómo agregar desencadenadores de restauración durante la creación o actualización de una pila (AWS CLI)

Utilice el comando [create-stack](#) o [update-stack](#) con la opción `--rollback-configuration`.

Por ejemplo, el comando `update-stack` siguiente define *MyCompositeAlarm* como desencadenador de restauración con un periodo de supervisión de 5 minutos:

```
aws cloudformation update-stack --stack-name MyStack \  
  --use-previous-template \  
  --rollback-configuration \  
    "RollbackTriggers=[{Arn=arn:aws:cloudwatch:us-  
east-1:123456789012:alarm:MyCompositeAlarm,Type=AWS::CloudWatch::CompositeAlarm}],MonitoringTim
```

Agregar disparadores de restauración a un conjunto de cambios

Cómo agregar desencadenadores de restauración a un conjunto de cambios (consola)

1. Durante la creación de un conjunto de cambios, en la página Configurar opciones de pila, en Opciones avanzadas, amplíe la sección Configuración de restauración.
2. Ingrese un tiempo de supervisión comprendido entre 0 y 180 minutos. El valor predeterminado es 5.
3. Especifique el ARN de la alarma o la alarma compuesta de CloudWatch que desee utilizar como desencadenador de restauración y seleccione Agregar el ARN de alarma de CloudWatch.

Por ejemplo, el siguiente es un ARN para una alarma de CloudWatch o una alarma compuesta, `arn:aws:cloudwatch:us-east-1:123456789012:alarm:MyAlarmName`.

4. Seleccione Siguiente y revise los detalles de la pila.
5. Cuando tenga todo listo, seleccione Crear conjunto de cambios para crear el conjunto de cambios.

Cómo agregar desencadenadores de restauración a un conjunto de cambios (AWS CLI)

Utilice el comando [create-change-set](#) con la opción `--rollback-configuration`.

Ver los disparadores de restauración de una pila

Para ver los desencadenadores de restauración de una pila, consulte la sección Rollback configuration (Configuración de restauración).

1. En la página Pilas, seleccione la pila que desea ver de la lista de la izquierda.
2. En la pestaña Información de la pila, amplíe la sección Configuración de restauración para ver los desencadenadores de restauración.

Ver desencadenadores de restauración para un conjunto de cambios

Para ver los desencadenadores de restauración de un conjunto de cambios, consulte la sección Rollback configuration (Configuración de restauración).

1. En la página Pilas, seleccione la pila que desea ver de la lista de la izquierda.
2. Seleccione la pestaña Conjuntos de cambios y, a continuación, seleccione el conjunto de cambios que desee ver.
3. Seleccione la pestaña Entrada y consulte la sección Configuración de restauración.

Detectar cambios de configuración no administrados en pilas y recursos con detección de derivación

Incluso aunque administre sus recursos a través de CloudFormation, los usuarios pueden cambiar esos recursos desde fuera de CloudFormation. Los usuarios pueden editar los recursos directamente usando el servicio subyacente que ha creado el recurso. Por ejemplo, puede utilizar la consola de Amazon EC2 para actualizar una instancia de servidor que se creó como parte de una pila de CloudFormation. Algunos cambios pueden ser accidentales, mientras que otros pueden realizarse a propósito para dar respuesta a eventos operativos en los que el tiempo es crucial. Sea cual sea el motivo, los cambios realizados fuera de CloudFormation pueden complicar las operaciones de actualización de la pila o de eliminación. Puede utilizar la detección de desviaciones para identificar los recursos de pila en los que se han introducido cambios de configuración fuera de la administración de CloudFormation. Después podrá adoptar las medidas correctivas adecuadas para que los recursos de la pila vuelvan a estar sincronizados con sus definiciones en la plantilla de pila; por ejemplo, puede actualizar los recursos con desviaciones directamente para que concuerden con su definición en la plantilla. Solucionar las desviaciones ayuda a garantizar la coherencia de la configuración y la correcta ejecución de las operaciones de pila.

Temas

- [¿Qué es una desviación?](#)
- [Códigos de estado de detección de desviaciones](#)
- [Consideraciones a tener en cuenta cuando se detectan desviaciones](#)
- [Detección de desviaciones en una pila de CloudFormation completa](#)
- [Detección de desviaciones en recursos de la pila individuales](#)
- [Resolver desviaciones con una operación de importación](#)

¿Qué es una desviación?

La detección de desviaciones le permite detectar si la configuración real de la pila es diferente o se ha desviado de su configuración esperada. CloudFormation se utiliza para detectar las desviaciones en una pila completa o en recursos individuales dentro de la pila. Se considera que un recurso se ha desviado si alguno de sus valores de propiedad reales son diferentes de los valores de propiedad esperados. Esto incluye la eliminación de la propiedad o el recurso. Se considera que una pila se ha desviado si uno o varios de sus recursos se han desviado.

Con el fin de determinar si un recurso se ha desviado, CloudFormation determina los valores de propiedades de recursos esperados, tal y como están definidos en la plantilla de la pila y los valores especificados como los parámetros de la plantilla. A continuación, CloudFormation compara los valores esperados con los valores reales de esas propiedades de recursos, tal y como son actualmente en la pila. Se considera que un recurso se ha desviado si una o varias de sus propiedades se han eliminado o si el valor de las propiedades ha cambiado.

CloudFormation genera información detallada sobre cada recurso de la pila que se ha desviado.

CloudFormation detecta las desviaciones en los recursos de AWS que admiten la detección de desviaciones. A los recursos que no admiten dicha detección se les asigna un estado de desviación NOT_CHECKED. Para obtener una lista de AWS los recursos que admiten la detección de desviaciones, consulte [Tipo de recurso compatible](#).

Además, CloudFormation admite detección de desviaciones en tipos de recursos privados que son provisionables; es decir, cuyo tipo de aprovisionamiento es FULLY_MUTABLE o IMMUTABLE. Para realizar la detección de desviaciones en un recurso de tipo privado, la versión predeterminada del tipo de recurso que ha registrado en la cuenta debe ser provisionable. Para obtener más información sobre tipos de aprovisionamiento de recursos, consulte el parámetro ProvisioningType de la acción [DescribeType](#) en la Referencia API de AWS CloudFormation y del comando [DescribeType](#) en

la Referencia del comando AWS CLI. Para obtener más información acerca de los recursos privados, consulte [Administración de extensiones a través del registro de CloudFormation](#).

Puede ejecutar la detección de desviaciones en pilas que tengan los siguientes estados: `CREATE_COMPLETE`, `UPDATE_COMPLETE`, `UPDATE_ROLLBACK_COMPLETE` y `UPDATE_ROLLBACK_FAILED`.

Cuando detecta desviaciones en una pila, CloudFormation no detecta desviaciones en ninguna de las pilas anidadas que pertenecen a dicha pila. Para obtener más información, consulte [División de una plantilla en piezas reutilizables utilizando pilas anidadas](#). En su lugar, puede iniciar una operación de detección de desviaciones directamente en la pila anidada.

Note

CloudFormation determina únicamente las desviaciones de los valores de propiedades que se establecen explícitamente, ya sea a través de la plantilla de pila o especificando los parámetros de plantilla. No se incluyen los valores predeterminados de las propiedades de los recursos. Para que CloudFormation realice el seguimiento de una propiedad de recurso para determinar la desviación, defina de forma explícita el valor de la propiedad, aunque lo configure en el valor predeterminado.

Códigos de estado de detección de desviaciones

En las tablas de esta sección se describen los distintos tipos de estado que se utilizan para la detección de desviaciones:

- En Estado de operación de detección de desviación se describe el estado actual de la operación de desviación.
- Estado de desviación

Para los conjuntos de pila, esto describe el estado de desviación del conjunto de pila en su totalidad, en función del estado de desviación de las instancias de pila que le pertenecen.

Para las instancias de pila, esto describe el estado de desviación de la instancia de pila, en función del estado de desviación de su pila asociada.

Para las pilas, esto describe el estado de desviación de la pila en su conjunto, en función del estado de desviación de sus recursos.

- En Estado de desviación del recurso se describe el estado de desviación de un recurso individual.

En la siguiente tabla se enumeran los códigos de estado que CloudFormation asigna a las operaciones de detección de desviaciones de pila.

Estado de la operación de detección de desviaciones	Descripción
DETECTION_COMPLETE	La operación de detección de desviaciones de la pila se ha realizado con éxito para todos los recursos de la pila que admiten la detección de desviaciones.
DETECTION_FAILED	La operación de detección de desviaciones de la pila ha generado un error para un recurso de la pila como mínimo. Los resultados estarán disponibles para los recursos en los que CloudFormation ha completado correctamente la detección de desviaciones.
DETECTION_IN_PROGRESS	La operación de detección de desviaciones de la pila está en curso actualmente.

En la siguiente tabla se enumeran los códigos de estado de desviación que CloudFormation asigna a las pilas.

Estado de desviación	Descripción
DRIFTED	Para pilas: la pila difiere o se ha desviado de su configuración de plantilla prevista. Se considera que una pila se ha desviado si uno o varios de sus recursos se han desviado. Para instancias de pila: se considera que una instancia de pila se ha desviado si la pila asociada con ella se ha desviado. Para conjuntos de pila: se considera que un conjunto de pila se ha desviado si una o más instancias de pila se han desviado.

Estado de desviación	Descripción
NOT_CHECKED	CloudFormation no ha comprobado si la pila, el conjunto de pila o la instancia de pila difiere de su configuración de plantilla esperada.
IN_SYNC	La configuración actual de cada recurso admitido coincide con su configuración de plantilla prevista. Una pila, un conjunto de pila o una instancia de pila sin recursos que admitan la detección de desviación también tendrá un estado de IN_SYNC.

En la siguiente tabla se enumeran los códigos de estado de desviación que CloudFormation asigna a los recursos de pila.

Estado de desviación de recursos	Descripción
DELETED	El recurso difiere de la configuración de plantilla prevista ya que se ha eliminado el recurso.
MODIFIED	El recurso difiere de su configuración de plantilla prevista.
NOT_CHECKED	CloudFormation no ha comprobado si el recurso difiere de su configuración de plantilla prevista.
IN_SYNC	La configuración actual del recurso coincide con su configuración de plantilla prevista.

En la tabla siguiente se muestran los códigos de estado del tipo diferencia que CloudFormation asigna a las propiedades de los recursos que difieren de su configuración de plantilla prevista.

Tipos de diferencia de propiedad	Descripción
ADD	Se ha agregado un valor a una propiedad de recurso que es un tipo de datos lista o matriz.

Tipos de diferencia de propiedad	Descripción
REMOVE	La propiedad se ha eliminado de la configuración de recursos actual.
NOT_EQUAL	El valor de propiedad actual difiere de su valor esperado tal y como se define en la plantilla de pila.

Consideraciones a tener en cuenta cuando se detectan desviaciones

Con el fin de realizar correctamente una detección de desviaciones en una pila, el usuario debe tener los siguientes permisos:

- Permiso de lectura para cada recurso que admite la detección de desviaciones incluida en la pila. Por ejemplo, si la pila incluye un recurso `AWS::EC2::Instance`, debe disponer de permiso `ec2:DescribeInstances` para realizar una detección de desviaciones en la pila.
- `cloudformation:DetectStackDrift`
- `cloudformation:DetectStackResourceDrift`
- `cloudformation:BatchDescribeTypeConfigurations`

Para más información acerca de cómo establecer permisos en CloudFormation, consulte [Control del acceso a CloudFormation con AWS Identity and Access Management](#).

En algunos casos de borde, es posible que CloudFormation no pueda devolver siempre resultados de desviaciones precisos. Debe tener en cuenta estos casos de límite para interpretar correctamente sus resultados de detección de desviaciones.

- En algunos casos los objetos contenidos en matrices de propiedades se indican como desviaciones, cuando en realidad son valores predeterminados suministrados a la propiedad desde el servicio subyacente responsable del recurso.
- Algunos recursos tienen relaciones de asociación con recursos relacionados, de forma que un recurso en realidad pueda asociar o eliminar valores de otro recurso que esté definido en la misma plantilla o en otra. Por ejemplo, los recursos `AWS::EC2::SecurityGroupIngress` y `AWS::EC2::SecurityGroupEgress` se pueden utilizar para asociar y eliminar valores de recursos `AWS::EC2::SecurityGroup`. En estos casos CloudFormation analiza la plantilla de la pila para detectar asociaciones antes de realizar la comparación de desviación. Sin embargo,

CloudFormation no puede realizar este análisis en varias pilas, por lo que es posible que no devuelva resultados precisos de desviaciones cuando los recursos asociados residan en diferentes pilas.

Los recursos que admiten detección de desviaciones y permiten o requieren asociaciones de otros recursos son:

Tipo de recurso	Tipo de recurso de asociación
<code>AWS::SNS::Topic</code>	<code>AWS::SNS::Subscription</code>
<code>AWS::IAM::User</code>	<code>AWS::IAM::UserToGroupAddition</code>
<code>AWS::IAM::Group</code>	<code>AWS::IAM::ManagedPolicy</code>
<code>AWS::IAM::Role</code>	<code>AWS::IAM::Policy</code>
<code>AWS::IAM::User</code>	<code>AWS::IAM::RolePolicy</code>
<code>AWS::ElasticLoadBalancingV2::Listener</code>	<code>AWS::ElasticLoadBalancingV2::ListenerCertificate</code>
<code>AWS::EC2::SecurityGroup</code>	<code>AWS::EC2::SecurityGroupEgress</code> <code>AWS::EC2::SecurityGroupIngress</code>

- CloudFormation no realiza una detección de desviaciones en la propiedad `KMSKeyId` de los recursos. Dado que varios alias pueden hacer referencia a las claves de AWS KMS, CloudFormation no puede garantizar resultados de desviaciones precisos coherentes para esta propiedad.
- Existen algunas propiedades de recursos que puede especificar en su plantilla de pila y que, debido a su propia naturaleza, CloudFormation no puede comparar con las propiedades de los recursos de pila obtenidos. Por lo tanto, estas propiedades no se pueden incluir en los resultados de detección de las desviaciones. Dichas propiedades se dividen en dos amplias categorías:
 - Valores de propiedades que CloudFormation no puede asignar de nuevo a su valor inicial de propiedad de recurso en la plantilla de la pila.

Por ejemplo, CloudFormation no puede volver a asignar un código fuente de una función de Lambda al tipo de propiedad [Code](#) del recurso [AWS::Lambda::Function](#) y, por lo tanto, CloudFormation no puede incluirlo en los resultados de detección de desviaciones.

- Los valores de propiedades que el servicio responsable del recurso no devuelve.

Existen determinados valores de propiedades que, por diseño, el servicio al que pertenece el recurso nunca devuelve. Se trata de valores que tienden a contener información confidencial, como contraseñas u otro tipo de información confidencial, que no debe estar expuesta.

Por ejemplo, el servicio IAM nunca devuelve el valor de la propiedad Password del tipo de propiedad [AWS::IAM::User LoginProfile](#) de usuario de IAM y, por lo tanto, CloudFormation no puede incluirlo en los resultados de detección de desviaciones.

- Es posible que los objetos de una matriz sean en realidad valores predeterminados del servicio y no desviaciones agregadas manualmente.
- Si encuentra algún falso positivo, envíenos sus comentarios mediante el enlace de comentarios de la consola de CloudFormation o póngase en contacto con nosotros a través de [AWS re:Post](#).
- Algunas propiedades pueden tener valores de entrada iguales, pero no idénticos. Para evitar falsos positivos, debe asegurarse de que la configuración esperada coincida con la configuración real.
- Por ejemplo, la configuración esperada de la propiedad de recurso puede ser de 1024 MB y la configuración real de la misma propiedad de recurso puede ser de 1 GB. 1024 MB y 1 GB son iguales, pero no idénticos.

Cuando se ejecuta la detección de desviaciones en esta propiedad de recurso, indicará los resultados de las desviaciones.

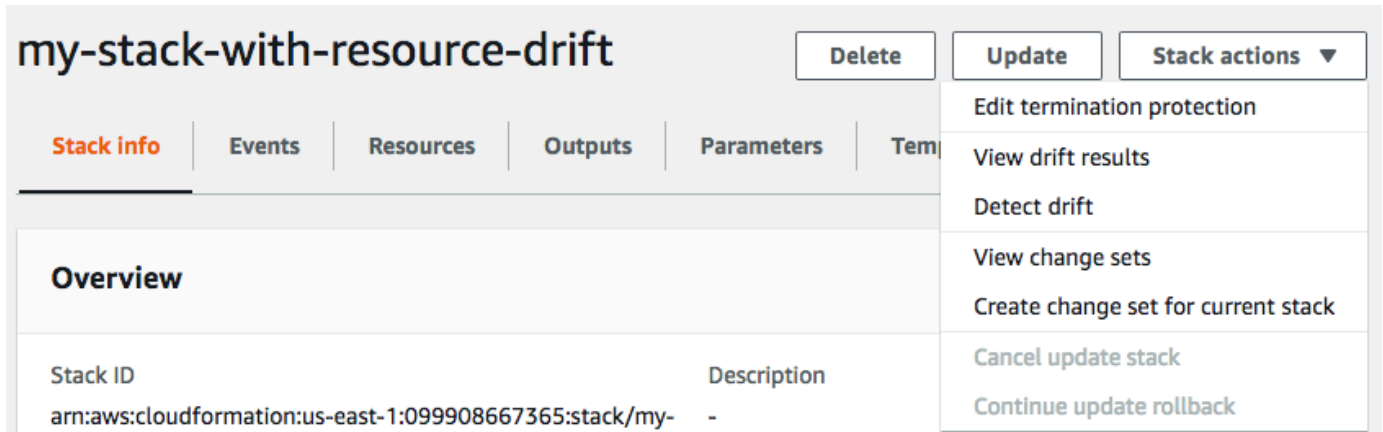
Para evitar este falso positivo, cambie la configuración esperada de la propiedad del recurso a 1024 MB y, a continuación, ejecute la detección de desviaciones.

Detección de desviaciones en una pila de CloudFormation completa

La realización de una operación de detección de desviaciones en una pila determina si la pila se ha desviado de su configuración de plantilla prevista y permite devolver información detallada sobre el estado de la desviación de cada recurso de la pila que admite la detección de desviaciones.

Para detectar desviaciones en una pila completa con la Consola de administración de AWS

1. Abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation>.
2. En la lista de pilas, seleccione la pila en la que desea detectar desviaciones. En el panel de detalles de la pila, elija Stack actions (Acciones de pila) y, a continuación, seleccione Detect drift (Detectar desviación).



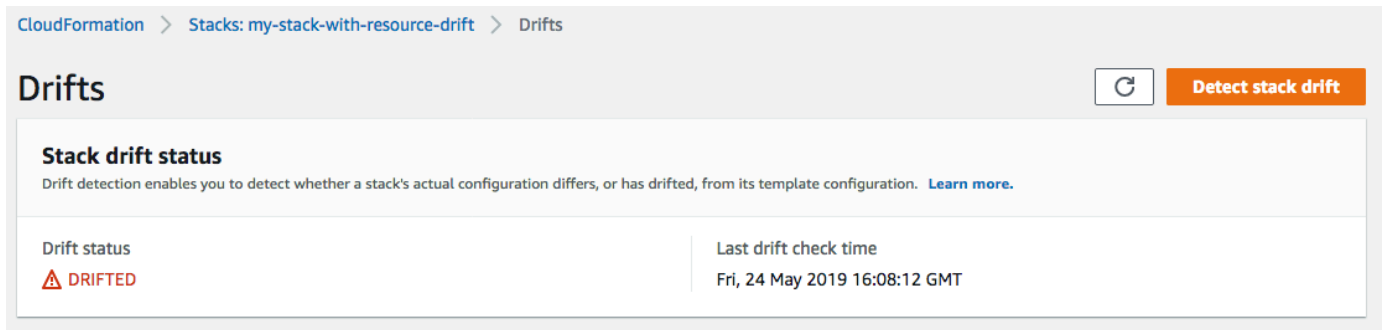
CloudFormation muestra una barra de información que indica que se ha iniciado la detección de desviaciones en la pila seleccionada.

3. Espere hasta que CloudFormation complete la operación de detección de desviaciones. Cuando la operación de detección de desviaciones finaliza, CloudFormation actualiza Drift status (Estado de desviaciones) y Last drift check time (Último momento de comprobación de desviaciones) para la pila. Estos campos se enumeran en la sección Overview (Información general) del panel Stack info (Información de pila) de la página de detalles de pila.

La operación de detección de desviaciones puede tardar varios minutos, según la cantidad de recursos que contiene la pila. Solo puede ejecutar una única operación de detección de desviaciones en una pila determinada al mismo tiempo. CloudFormation continúa con la operación de detección de desviaciones incluso después de cerrar la barra de información.

4. Revise los resultados de la detección de desviaciones de la pila y sus recursos. Con la pila seleccionada, desde el menú desviaciones de acciones Stack actions (Acciones de pila), seleccione View drift results (Ver resultados de desviaciones).

CloudFormation muestra el estado de las desviaciones en general de la pila, además de la última vez en que se inició la detección de desviaciones en la pila o en cualquiera de sus recursos individuales. Se considera que una pila se ha desviado si uno o varios de sus recursos se han desviado.



En la sección Resource drift details (Detalles de las desviaciones de recursos), CloudFormation genera una lista de cada recurso de pila, su estado de desviación y la última vez que se inició la detección de desviaciones en ese recurso. El ID lógico y el ID físico de cada recurso se muestra para ayudarle a identificarlos. Además, en el caso de los recursos con el estado MODIFIED, CloudFormation muestra los detalles de la desviación del recurso.

Puede ordenar los recursos en función de su estado mediante la columna Drift status (Estado de desviación).

- Para ver los detalles en un recurso modificado.
 - Con el recurso modificado seleccionado, elija View drift details (Ver detalles de desviación).

CloudFormation muestra la página de detalles de desviación de ese recurso. La página muestra los valores de propiedades actuales y previstos del recurso, así como las diferencias entre los dos.

Para resaltar una diferencia, en la sección Differences (Diferencias), seleccione el nombre de la propiedad.

- Las propiedades agregadas se resaltan en verde en la columna Current (Actual) de la sección Details (Detalles).
- Las propiedades eliminadas están marcadas en rojo en la columna Expected (Previsto) de la sección Details (Detalles).
- Las propiedades cuyo valor ha cambiado aparecen resaltadas en amarillo en las columnas Previsto y Current (Actual).

Differences (4)				
☒	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

Actual

```
{
  "AttributeDefinitions": [
    {
      "AttributeName": "Name",
      "AttributeType": "S"
    },
  ],
  "KeySchema": [
    {
      "AttributeName": "Name",
      "KeyType": "HASH"
    }
  ],
  "ProvisionedThroughput": {
    "ReadCapacityUnits": 5,
    "WriteCapacityUnits": 5
  },
  "Tags": [
    {
      "Key": "test",
      "Value": "test"
    }
  ]
}
```

```
{
  "AttributeDefinitions": [
    {
      "AttributeName": "Name",
      "AttributeType": "S"
    },
    {
      "AttributeName": "stuff",
      "AttributeType": "S"
    }
  ],
  "KeySchema": [
    {
      "AttributeName": "Name",
      "KeyType": "HASH"
    }
  ],
  "ProvisionedThroughput": {
    "ReadCapacityUnits": 6,
    "WriteCapacityUnits": 4
  }
}
```

Para detectar desviaciones en una pila completa con la AWS CLI

 Important

Revise la Hora de la última verificación de desviación de la pila y confirme que sea anterior a la marca de tiempo que aparece en los resultados de desviación del recurso, para evitar el uso de datos obsoletos.

Para detectar desviaciones en una pila completa a través de la AWS CLI, utilice los siguientes comandos de AWS CLI:

- `detect-stack-drift` para iniciar una operación de detección de desviaciones en una pila.
- `describe-stack-drift-detection-status` para monitorizar el estado de operación de detección de desviaciones de la pila.
- `describe-stack-resource-drifts` para revisar los detalles de la operación de detección de desviaciones de la pila.

1. Use la `detect-stack-drift` para detectar desviaciones en una pila completa. Especifique el nombre de pila o ARN. También puede especificar los ID lógicos de los recursos específicos que desee utilizar como filtros para esta operación de detección de desviaciones de filtros.

```
aws cloudformation detect-stack-drift --stack-name my-stack-with-resource-drift
```

Salida:

```
{
  "StackDriftDetectionId": "624af370-311a-11e8-b6b7-500cexample"
}
```

2. Debido a que las operaciones de detección de desviaciones de pila pueden mantenerse durante mucho tiempo, utilice `describe-stack-drift-detection-status` para monitorizar el estado de operación de desviaciones. Este comando toma el ID de detección de desviaciones de la pila que el comando `detect-stack-drift` devuelve.

En el ejemplo siguiente hemos tomado el ID de detección de desviaciones de la pila devuelto por el ejemplo de `detect-stack-drift` anterior y lo hemos transferido a `describe-stack-drift-detection-status`. El parámetro devuelve detalles de operación que muestran que la operación de detección de desviaciones se ha completado, un solo recurso de la pila se ha desviado y que, por tanto, se considera que toda la pila se ha desviado.

```
aws cloudformation describe-stack-drift-detection-status --stack-drift-detection-id 624af370-311a-11e8-b6b7-500cexample
```

Salida:

```
{
```

```

    "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. Cuando la operación de detección de desviaciones de la pila se haya completado, utilice el comando `describe-stack-resource-drifts` para revisar los resultados, incluidos los valores de propiedades reales y previstos para los recursos que se han desviado.

En el ejemplo siguiente se utiliza la opción `--stack-resource-drift-status-filters` para pedir información sobre las desviaciones de la pila para los recursos que se han modificado o se han eliminado. La solicitud devuelve información sobre el recurso que se ha modificado, como los detalles sobre dos de sus propiedades cuyos valores se han cambiado. No se ha eliminado ningún recurso.

```

aws cloudformation describe-stack-resource-drifts --stack-name my-stack-with-
resource-drift --stack-resource-drift-status-filters MODIFIED DELETED

```

Salida:

```

{
  "StackResourceDrifts": [
    {
      "StackId": "arn:aws:cloudformation:us-east-1:099908667365:stack/my-
stack-with-resource-drift/489e5570-df85-11e7-a7d9-50example",
      "ActualProperties": "{\n\"ReceiveMessageWaitTimeSeconds\":0,
\n\"DelaySeconds\":120,\n\"RedrivePolicy\":{\n\"deadLetterTargetArn\":\n\"arn:aws:sqs:us-
east-1:099908667365:my-stack-with-resource-drift-DLQ-1BCY7HHD5QIM3\",
\n\"maxReceiveCount\":12},\n\"MessageRetentionPeriod\":345600,\n\"MaximumMessageSize
\n\":262144,\n\"VisibilityTimeout\":60,\n\"QueueName\":\n\"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"
  }
]
}

```

Detección de desviaciones en recursos de la pila individuales

Puede detectar desviaciones en recursos específicos de una pila, en lugar de toda la pila. Esto resulta especialmente útil cuando solo necesita determinar si ahora recursos específicos vuelven a coincidir con sus configuraciones de plantilla previstas.

Al realizar la detección de desviaciones en un recurso, CloudFormation también actualiza el estado de las desviaciones de la pila en general y la última hora de comprobación de las desviaciones, si es aplicable. Por ejemplo, supongamos que una pila tiene un estado de desviación de `IN_SYNC`. Hace que CloudFormation detecte desviaciones en uno o varios recursos contenidos en dicha pila y CloudFormation detecta que uno o varios de esos recursos se han desviado. CloudFormation actualiza el estado de desviación de la pila a `DRIFTED`. Por el contrario, supongamos que tiene una pila con un estado de desviaciones `DRIFTED` debido a un único recurso desviado. Si vuelve a establecer dicho recurso en sus valores de propiedades previstos y después vuelve a detectar desviaciones en dicho recurso, CloudFormation actualizará el estado de desviación de recurso y el

estado de desviación de pila a IN_SYNC sin que sea preciso detectar la desviación en toda la pila de nuevo.

Para detectar desviaciones en un recurso individual con la Consola de administración de AWS

1. Abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation>.
2. En la lista de pilas, seleccione la pila que contiene el recurso. CloudFormation muestra los detalles de la pila.
3. En el panel de navegación izquierdo, en Pilas, elija Acciones de pila y, a continuación, elija Detectar desviación.
4. En Resource drift status (Estado de desviación de recurso), elija el recurso y, a continuación, seleccione Detect drift for resource (Detectar desviación de recurso).

CloudFormation realiza la detección de desviaciones en el recurso seleccionado. Si se ejecuta correctamente, CloudFormation actualiza el estado de la desviación del recurso y el estado de la desviación de la pila en general, si es necesario. CloudFormation también actualiza la marca temporal de la última vez que se ejecutó una detección de desviaciones en el recurso y la pila en su conjunto. Si el recurso se ha modificado, CloudFormation muestra información detallada de la desviación de los valores de propiedades actuales y previstos del recurso.

5. Revise los resultados de la detección de desviaciones del recurso.
 - Para ver los detalles en un recurso modificado.
 - Con el recurso modificado seleccionado, elija View drift details (Ver detalles de desviación).

CloudFormation muestra los detalles de desviación del recurso, incluidos los valores de las propiedades actuales y previstas del recurso y las diferencias entre los dos.

Para resaltar una diferencia, en la sección Differences (Diferencias), seleccione el nombre de la propiedad.

- Las propiedades agregadas se resaltan en verde en la columna Current (Actual) de la sección Details (Detalles).
- Las propiedades eliminadas están marcadas en rojo en la columna Expected (Previsto) de la sección Details (Detalles).
- Las propiedades cuyo valor ha cambiado aparecen resaltadas en amarillo en las columnas Previsto y Current (Actual).

Differences (4)				
☒	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

Actual

```
{
  "AttributeDefinitions": [
    {
      "AttributeName": "Name",
      "AttributeType": "S"
    },
  ],
  "KeySchema": [
    {
      "AttributeName": "Name",
      "KeyType": "HASH"
    }
  ],
  "ProvisionedThroughput": {
    "ReadCapacityUnits": 5,
    "WriteCapacityUnits": 5
  },
  "Tags": [
    {
      "Key": "test",
      "Value": "test"
    }
  ]
}
```

```
{
  "AttributeDefinitions": [
    {
      "AttributeName": "Name",
      "AttributeType": "S"
    },
    {
      "AttributeName": "stuff",
      "AttributeType": "S"
    }
  ],
  "KeySchema": [
    {
      "AttributeName": "Name",
      "KeyType": "HASH"
    }
  ],
  "ProvisionedThroughput": {
    "ReadCapacityUnits": 6,
    "WriteCapacityUnits": 4
  }
}
```

Para detectar desviaciones en un recurso individual con la AWS CLI

- 

Important

Revise la Hora de la última verificación de desviación del recurso de pila y confirme que sea anterior a la marca de tiempo que aparece en los resultados de desviación del recurso, para evitar el uso de datos obsoletos.

Para detectar desviaciones en un recurso individual con la AWS CLI, ejecute el comando `detect-stack-resource-drift`. Especifique el ID lógico del recurso, así como la pila en la que se encuentra.

En el siguiente ejemplo se ejecuta una operación de detección de desviaciones en recursos específicos de una pila, `my-drifted-resource`. La respuesta devuelve información que confirma que el recurso se ha modificado, como los detalles sobre dos de sus propiedades cuyos valores se han cambiado.

```
aws cloudformation detect-stack-resource-drift \
  --stack-name my-stack-with-resource-drift \
  --logical-resource-id my-drifted-resource
```

Salida:

```
{
  "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"
  }
}
```

Resolver desviaciones con una operación de importación

Puede haber casos en los que la configuración de un recurso se haya desviado de la configuración prevista y desee aceptar la nueva configuración como la configuración prevista. En la mayoría de los casos, se resolverían los resultados de desviación actualizando la definición de recursos en la plantilla de pila con una nueva configuración y, a continuación, se realizaría una actualización de pila. Sin embargo, si la nueva configuración actualiza una propiedad de recurso que requiere reemplazo, el recurso se volverá a crear durante la actualización de la pila. Si desea retener el recurso existente, puede utilizar la característica de importación de recursos para actualizarlo y resolver los resultados de desviación sin hacer que se reemplace el recurso.

La resolución de desviaciones de un recurso mediante una operación de importación consta de los siguientes pasos básicos:

- [Agregue un atributo `DeletionPolicy`, establecido en `Retener`, al recurso](#). Esto garantiza que se retenga el recurso existente en lugar de eliminarlo cuando se quita de la pila.
- [Elimine el recurso de la plantilla y ejecute una operación de actualización de la pila](#). Esto quita el recurso de la pila, pero no lo elimina.
- [Describa el estado real del recurso en la plantilla de la pila y, a continuación, importe el recurso existente de nuevo a la pila](#). Esto agrega el recurso de nuevo en la pila y resuelve las diferencias de propiedad que estaban provocando los resultados de desviación.

Para obtener más información sobre la importación de recursos, consulte [Cómo importar recursos de AWS a una pila de CloudFormation de forma manual](#). Para obtener una lista de los recursos que admiten operaciones de importación, consulte [Tipo de recurso compatible](#).

En este ejemplo, utilizamos la siguiente plantilla, denominada `templateToImport.json`.

Example JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "Import test",
  "Resources": {
    "ServiceTable": {
      "Type": "AWS::DynamoDB::Table",
      "Properties": {
        "TableName": "Service",
        "AttributeDefinitions": [
          {
            "AttributeName": "key",
            "AttributeType": "S"
          }
        ],
        "KeySchema": [
          {
            "AttributeName": "key",
            "KeyType": "HASH"
          }
        ],
        "BillingMode": "PROVISIONED",
        "ProvisionedThroughput": {
          "ReadCapacityUnits": 5,
          "WriteCapacityUnits": 1
        }
      }
    },
    "GamesTable": {
      "Type": "AWS::DynamoDB::Table",
      "Properties": {
        "TableName": "Games",
        "AttributeDefinitions": [
          {
            "AttributeName": "key",
            "AttributeType": "S"
          }
        ],
        "KeySchema": [
          {
            "AttributeName": "key",
```

```

        "KeyType": "HASH"
      }
    ],
    "BillingMode": "PROVISIONED",
    "ProvisionedThroughput": {
      "ReadCapacityUnits": 5,
      "WriteCapacityUnits": 1
    }
  }
}
}
}
}

```

Example YAML

```

AWSTemplateFormatVersion: 2010-09-09
Description: Import test
Resources:
  ServiceTable:
    Type: AWS::DynamoDB::Table
    Properties:
      TableName: Service
      AttributeDefinitions:
        - AttributeName: key
          AttributeType: S
      KeySchema:
        - AttributeName: key
          KeyType: HASH
      BillingMode: PROVISIONED
      ProvisionedThroughput:
        ReadCapacityUnits: 5
        WriteCapacityUnits: 1
  GamesTable:
    Type: AWS::DynamoDB::Table
    Properties:
      TableName: Games
      AttributeDefinitions:
        - AttributeName: key
          AttributeType: S
      KeySchema:
        - AttributeName: key
          KeyType: HASH
      BillingMode: PROVISIONED

```

ProvisionedThroughput:
ReadCapacityUnits: 5
WriteCapacityUnits: 1

En este ejemplo, supongamos que un usuario ha cambiado un recurso fuera de CloudFormation. Después de ejecutar la detección de desviaciones, descubrimos que en GamesTable BillingMode se ha modificado por PAY_PER_REQUEST. Para obtener más información sobre la detección de desviaciones, consulte [Detectar cambios de configuración no administrados en pilas y recursos con detección de derivación](#).

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

```
{  "AttributeDefinitions": [    {      "AttributeName": "key",      "AttributeType": "S"    }  ],  "BillingMode": "PROVISIONED",  "KeySchema": [    {      "AttributeName": "key",      "KeyType": "HASH"    }  ],  "ProvisionedThroughput": {    "ReadCapacityUnits": 5,    "WriteCapacityUnits": 1  },  "TableName": "Games" }
```

Actual

```
{  "AttributeDefinitions": [    {      "AttributeName": "key",      "AttributeType": "S"    }  ],  "BillingMode": "PAY_PER_REQUEST",  "KeySchema": [    {      "AttributeName": "key",      "KeyType": "HASH"    }  ],  "ProvisionedThroughput": {    "ReadCapacityUnits": 0,    "WriteCapacityUnits": 0  },  "TableName": "Games" }
```

Nuestra pila no está actualizada, nuestros recursos están activos, pero queremos preservar la configuración de recursos prevista. Podemos hacer esto resolviendo la desviación mediante una operación de importación, sin interrumpir los servicios.

Resolver la desviación con una operación de importación mediante la consola de CloudFormation

Paso 1. Actualizar pila con la política de eliminación Retener

Para actualizar la pila mediante un atributo **DeletionPolicy** con la opción **Retain**

1. Inicie sesión en la Consola de administración de AWS y abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation>.
2. En la página Stacks (Pilas), elija la pila que se ha desviado.
3. Elija Update (Actualizar) y, a continuación, elija Replace current template (Reemplazar plantilla actual) del panel de detalles de la pila.

4. En la página Specify template (Especificar plantilla), proporcione la plantilla actualizada que contiene el atributo `DeletionPolicy` con la opción `Retain` mediante uno de los siguientes métodos:

- Seleccione URL de Amazon S3, y, a continuación, especifique la URL de la plantilla en el cuadro de texto.
- Seleccione Cargar un archivo de plantilla y, a continuación, busque la plantilla.

A continuación, elija Siguiente.

5. Revise la página Specify stack details (Especificar detalles de la pila) y elija Next (Siguiente).
6. Revise la página Configure stack options (Configurar opciones de la pila) y elija Next (Siguiente).
7. En la página Review ***stack-name*** (Revisar nombre de la pila), elija Update stack (Actualizar pila).

Resultados: en la página Events (Eventos) de la pila, el estado es `UPDATE_COMPLETE`.

Para resolver la desviación a través de una operación de importación, sin interrumpir los servicios, especifique una [DeletionPolicy](#) `Retain` para los recursos que desea eliminar de la pila. En el siguiente ejemplo, hemos agregado un atributo [DeletionPolicy](#), establecido en `Retain`, al recurso `GameTable`.

Example JSON

```
"GameTable": {
  "Type": "AWS::DynamoDB::Table",
  "DeletionPolicy": "Retain",
  "Properties": {
    "TableName": "Games",
```

Example YAML

```
GameTable:
  Type: AWS::DynamoDB::Table
  DeletionPolicy: Retain
  Properties:
    TableName: Games
```

Paso 2. Eliminar recursos desviados, parámetros relacionados y salidas

Para eliminar recursos desviados, parámetros relacionados y salidas

1. Elija Update (Actualizar) y, a continuación, elija Replace current template (Reemplazar plantilla actual) del panel de detalles de la pila.
2. En la página Specify template (Especificar plantilla), proporcione la plantilla actualizada con los recursos, parámetros relacionados y resultados eliminados de la plantilla de la pila mediante uno de los métodos siguientes:
 - Seleccione URL de Amazon S3, y, a continuación, especifique la URL de la plantilla en el cuadro de texto.
 - Seleccione Cargar un archivo de plantilla y, a continuación, busque la plantilla.

A continuación, elija Siguiente.

3. Revise la página Specify stack details (Especificar detalles de la pila) y elija Next (Siguiente).
4. Revise la página Configure stack options (Configurar opciones de la pila) y elija Next (Siguiente).
5. En la página Review **stack-name** (Revisar nombre de la pila), elija Update stack (Actualizar pila).

Resultados: Logical ID (ID lógico) GamesTable tiene un estado de DELETE_SKIPPED en la página Events (Eventos) de la pila.

Espere hasta que CloudFormation finalice la operación de actualización de la pila. Una vez finalizada la operación de actualización de la pila, elimine el recurso, los parámetros relacionados y los resultados de la plantilla de la pila. A continuación, importe la plantilla actualizada. Después de completar estas acciones, la plantilla de ejemplo tiene el siguiente aspecto.

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

```

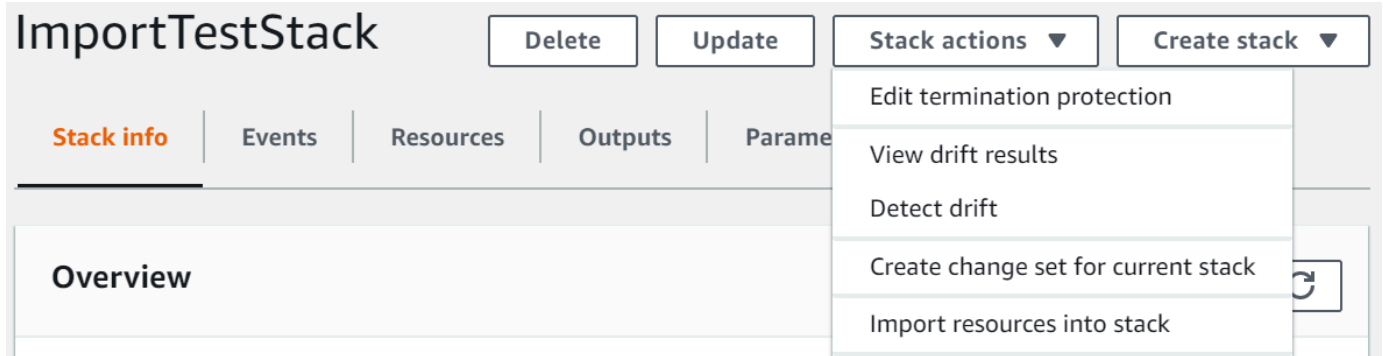
AWSTemplateFormatVersion: 2010-09-09
Description: Import test
Resources:
  ServiceTable:
    Type: AWS::DynamoDB::Table
    Properties:
      TableName: Service
      AttributeDefinitions:
        - AttributeName: key
          AttributeType: S
      KeySchema:
        - AttributeName: key
          KeyType: HASH
      BillingMode: PROVISIONED
      ProvisionedThroughput:
        ReadCapacityUnits: 5
        WriteCapacityUnits: 1

```

Paso 3. Actualizar la plantilla para que coincida con el estado activo de los recursos

Para actualizar la plantilla para que coincida con el estado activo de los recursos

1. Para importar la plantilla actualizada, elija Stack actions (Acciones de la pila) y, a continuación, elija Import resources into stack (Importar recursos en la pila).



2. Revise la página de Import overview (Información general sobre importación) para obtener una lista de los elementos que debe proporcionar durante esta operación y, a continuación, elija Next (Siguiente).
3. En la página Specify template (Especificar plantilla), proporcione la plantilla actualizada mediante uno de los métodos siguientes:
 - Seleccione URL de Amazon S3, y, a continuación, especifique la URL de la plantilla en el cuadro de texto.
 - Seleccione Cargar un archivo de plantilla y, a continuación, busque la plantilla.

A continuación, elija Siguiente.

4. En la página Identificar recursos, identifique cada recurso de destino. Para obtener más información, consulte [Identificadores de recursos](#).
 - a. En Propiedad del identificador, elija el tipo de identificador de recurso. Por ejemplo, la propiedad TableName identifica el recurso AWS::DynamoDB::Table.
 - b. En Identifier value (Valor del identificador), escriba el valor de la propiedad real. En la plantilla de ejemplo, TableName para el recurso GamesTable es Games.
 - c. Elija Siguiente.
5. Revise la página Specify stack details (Especificar detalles de pila) y elija Next (Siguiente).

6. En la página Import overview (Información general sobre la importación), revise los recursos importados y, a continuación, elija Import resources (Importar recursos). Esto importará el tipo de recurso `AWS::DynamoDB::Table` de nuevo a la pila.

Resultados: en este ejemplo, hemos resuelto la desviación de recursos mediante una operación de importación, sin interrumpir los servicios. Puede comprobar el progreso de una acción de importación en la consola de CloudFormation en la pestaña Eventos. Los recursos importados tendrán un estado de `IMPORT_COMPLETE` seguido de un estado de `CREATE_COMPLETE` con `Resource import complete` (Importación de recursos completada) como razón del estado.

Espere hasta que CloudFormation finalice la operación de actualización de la pila. Una vez finalizada la operación de actualización de la pila, la plantilla se actualiza para que coincida con el estado real desviado de los recursos. Por ejemplo, `BillingMode` se establecerá en `PAY_PER_REQUEST` y `ReadCapacityUnits` y `WriteCapacityUnits` se establecerán en `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
```

Cómo importar recursos de AWS a una pila de CloudFormation

Puede importar recursos existentes a una pila de CloudFormation. Esto es útil si desea comenzar a utilizar CloudFormation para administrar recursos creados fuera de CloudFormation, sin necesidad de eliminarlos ni recrearlos.

CloudFormation ofrece las siguientes opciones para importar recursos existentes a una pila:

- El [generador de laC](#) es una herramienta que escanea automáticamente los recursos existentes y genera una plantilla de CloudFormation en función de su estado actual. Luego, esta plantilla se puede usar para importar esos recursos a una pila.
- La [importación de recursos](#) es un proceso manual en el que se describen los recursos existentes en la plantilla de CloudFormation y, a continuación, se importan a una pila. Este enfoque requiere que especifique manualmente las propiedades y configuraciones de los recursos en la plantilla.
- La [importación automática](#) es un proceso automático en el que se describen los recursos existentes en la plantilla de CloudFormation, y CloudFormation importa aquellos con nombres personalizados coincidentes a una pila.

- La [refactorización de pilas](#) es una característica que simplifica la reorganización de los recursos de las pilas de CloudFormation y, al mismo tiempo, conserva las propiedades y los datos de los recursos existentes. Con la refactorización de pilas, puede mover recursos entre pilas, dividir pilas monolíticas en componentes más pequeños o consolidar varias pilas en una sola.

Además de poner los recursos existentes bajo la administración de CloudFormation, la característica de importación de recursos puede resultar útil en las siguientes situaciones:

- Mover recursos entre pilas: puede importar recursos de una pila a otra, lo que le permite reorganizar la infraestructura según sea necesario.
- Anidar las pilas existentes: puede importar una pila existente como una pila anidada dentro de otra pila, lo que permite diseños de infraestructura modulares y reutilizables.

CloudFormation es compatible con la importación de una amplia gama de recursos. Para obtener más información, consulte [Tipo de recurso compatible](#).

Temas

- [Cómo importar recursos de AWS a una pila de CloudFormation de forma manual](#)
- [Cómo importar recursos de AWS a una pila de CloudFormation automáticamente](#)
- [Reversión de una operación de importación](#)

Cómo importar recursos de AWS a una pila de CloudFormation de forma manual

Con la importación de recursos, puede importar recursos existentes de AWS a una pila nueva o existente de CloudFormation. Durante una operación de importación, se crea un conjunto de cambios que importa los recursos existentes en una pila o se crea una nueva pila a partir de los recursos existentes. Proporcione lo siguiente durante la importación.

- Una plantilla que describa toda la pila, incluidos los recursos de pila originales y los recursos que esté importando. Cada recurso que se va a importar debe tener un [atributo de DeletionPolicy](#).
- Identificadores de los recursos que está importando que CloudFormation puede usar para asignar los ID lógicos en la plantilla con los recursos existentes.

 Note

CloudFormation solo admite un nivel de anidamiento mediante la importación de recursos. Esto significa que no puede importar una pila a una pila secundaria ni importar una pila que tenga secundarias.

Temas

- [Identificadores de recursos](#)
- [Validación de la importación de recursos](#)
- [Códigos de estado de la importación de recursos](#)
- [Consideraciones durante una operación de importación](#)
- [Recursos adicionales de](#)
- [Creación de una pila a partir de recursos existentes](#)
- [Importación de recursos existentes en una pila](#)
- [Mover recursos entre pilas](#)
- [Anidamiento de una pila existente](#)

Identificadores de recursos

Proporcione dos valores para identificar cada recurso que esté importando.

- Una propiedad de identificador. Se trata de una propiedad de recurso que se puede utilizar para identificar cada tipo de recurso. Por ejemplo, un recurso `AWS::S3::Bucket` se puede identificar utilizando su `BucketName`.

La propiedad de recurso que utiliza para identificar el recurso que va a importar varía según el tipo de recurso. Puede encontrar la propiedad de recurso en la consola de CloudFormation. Tras crear una plantilla que incluya el recurso que se va a importar, puede iniciar el proceso de importación, donde encontrará las propiedades de identificadores de los recursos que va a importar. Para algunos tipos de recursos, puede haber varias formas de identificarlos y puede seleccionar qué propiedad usar en las listas desplegables.

Como alternativa, para obtener las propiedades de identificadores de los recursos que está importando, llame al comando de la CLI [get-template-summary](#) y especifique la URL de S3 de la plantilla de pila como valor de la opción `--template-url`.

- Un valor de identificador. Este es el valor real de la propiedad del recurso. Por ejemplo, el valor real de la propiedad `BucketName` podría ser `MyS3Bucket`.

Puede obtener el valor de la propiedad del identificador en la consola de servicio del recurso.

Validación de la importación de recursos

Durante una operación de importación, CloudFormation lleva a cabo las siguientes validaciones.

- El recurso que se va a importar existe.
- Las propiedades y los valores de configuración de cada recurso que se va a importar se adhieren al esquema de tipo de recurso, que define las propiedades aceptadas, las propiedades requeridas y los valores de propiedad admitidos.
- Las propiedades requeridas se especifican en la plantilla. Las propiedades requeridas para cada tipo de recurso se describen en la [Referencia de tipos de recursos y propiedades de AWS](#).
- El recurso que se va a importar no pertenece a otra pila de la misma región.

CloudFormation no comprueba si la configuración de la plantilla coincide con la configuración real de las propiedades del recurso.

Important

Compruebe que los recursos y las propiedades que se definieron en la plantilla coincidan con la configuración prevista de la importación de recursos para evitar cambios inesperados.

Códigos de estado de la importación de recursos

En esta tabla se describen los distintos tipos de estado utilizados con la característica de importación de recursos.

Estado de la operación de importación	Descripción
IMPORT_IN_PROGRESS	La operación de importación está en curso.

Estado de la operación de importación	Descripción
IMPORT_COMPLETE	La operación de importación se ha completado correctamente para todos los recursos de la pila.
IMPORT_ROLLBACK_IN_PROGRESS	La operación de restauración de importación ha restablecido la configuración de plantilla anterior.
IMPORT_ROLLBACK_FAILED	Se produjo un error en la operación de restauración de importación.
IMPORT_ROLLBACK_COMPLETE	La importación restableció la configuración de la plantilla anterior.

Consideraciones durante una operación de importación

- Una vez finalizada la importación y antes de realizar operaciones de pila posteriores, se recomienda ejecutar la detección de desviaciones en los recursos importados. La detección de desviaciones garantiza que la configuración de la plantilla coincida con la configuración real. Para obtener más información, consulte [Detección de desviaciones en una pila de CloudFormation completa](#).
- Las operaciones de importación no permiten nuevas creaciones de recursos, eliminaciones de recursos ni cambios en las configuraciones de propiedades.
- Cada recurso que se vaya a importar debe tener un atributo `DeletionPolicy` para que la operación de importación se realice correctamente. La `DeletionPolicy` se puede establecer en cualquier valor posible. Solo los recursos que está importando necesitan un parámetro `DeletionPolicy`. Los recursos que ya forman parte de la pila no necesitan una `DeletionPolicy`.
- No puede importar el mismo recurso en varias pilas.
- Puede utilizar la condición de política de IAM `cloudformation:ImportResourceTypes` para controlar con qué tipos de recursos pueden trabajar los usuarios durante una operación de importación. Para obtener más información, consulte [Claves de condición de políticas para CloudFormation](#).
- Al importar recursos se aplican los límites de pila de CloudFormation. Para obtener más información acerca los límites, consulte [Comprensión de las cuotas de CloudFormation](#).

Recursos adicionales de

Para solucionar el problema de la desviación de pilas mediante la importación de recursos, consulte [Resolver desviaciones con una operación de importación](#).

Creación de una pila a partir de recursos existentes

En este tema se muestra cómo crear una pila a partir de los recursos de AWS existentes al describirlos en una plantilla. Para buscar en su lugar los recursos existentes y generar automáticamente una plantilla que se pueda utilizar para importar los recursos existentes a CloudFormation o replicar los recursos en una cuenta nueva, consulte [Generar plantillas de los recursos existentes con el generador laC](#).

Requisitos previos

Antes de comenzar, debe disponer de lo siguiente:

- Una plantilla que describe todos los recursos que desea incluir en la pila nueva. Guarde la plantilla de manera local o en un bucket de Amazon S3.
- Para cada recurso que se desee importar, incluya lo siguiente:
 - Las propiedades y los valores de las propiedades que definen la configuración actual del recurso.
 - El identificador único por cada recurso; por ejemplo, por cada nombre. Para obtener más información, consulte [Identificadores de recursos](#).
 - El [atributo DeletionPolicy](#).

Temas

- [Plantilla de ejemplo](#)
- [Crear una pila a partir de recursos existentes mediante la Consola de administración de AWS](#)
- [Crear una pila a partir de recursos existentes mediante la AWS CLI](#)

Plantilla de ejemplo

En este tutorial, se supone que utiliza la siguiente plantilla de ejemplo, llamada `TemplateToImport.json`, la cual especifica dos tablas de DynamoDB que se crearon por fuera de CloudFormation. `ServiceTable` y `GameTable` son los destinos de la importación.

 Note

Esta plantilla es solo un ejemplo. Para utilizarla en las pruebas propias, sustituya los recursos de ejemplo con los recursos de su cuenta.

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



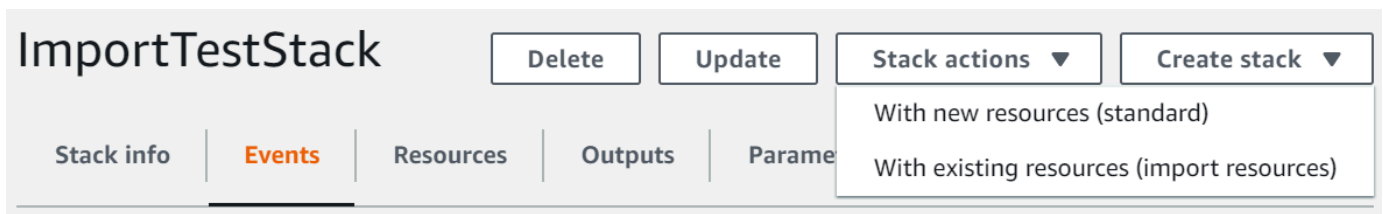
```

    },
    "KeySchema": [
      {
        "AttributeName": "key",
        "KeyType": "HASH"
      }
    ],
    "ProvisionedThroughput": {
      "ReadCapacityUnits": 5,
      "WriteCapacityUnits": 1
    }
  }
}

```

Crear una pila a partir de recursos existentes mediante la Consola de administración de AWS

1. Inicie sesión en la Consola de administración de AWS y abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation>.
2. En la página Pilas elija Crear pila, y, a continuación, elija Con recursos existentes (importar recursos).



3. Lea la página de información general sobre la importación para obtener una lista de los elementos que debe proporcionar durante esta operación. A continuación, elija Siguiente.
4. En la página Especificar plantilla proporcione la plantilla mediante uno de los métodos siguientes y, a continuación, elija Siguiente.
 - Seleccione URL de Amazon S3, y, a continuación, especifique la URL de la plantilla en el cuadro de texto.
 - Seleccione Cargar un archivo de plantilla y, a continuación, busque la plantilla.
5. En la página Identificar recursos, identifique cada recurso de destino. Para obtener más información, consulte [Identificadores de recursos](#).

- a. En Propiedad del identificador, elija el tipo de identificador de recurso. Por ejemplo, el recurso `AWS::DynamoDB::Table` se puede identificar mediante la propiedad `TableName`.
 - b. En Valor del identificador, escriba el valor de la propiedad real. Por ejemplo, el `TableName` para el recurso `GameTable` en el ejemplo de plantilla es *Games*.
 - c. Elija Siguiente.
6. En la página Especificar detalles de pila modifique los parámetros y, a continuación, elija Siguiente. Esto crea automáticamente un conjunto de cambios.

Important

La operación de importación produce un error si modifica parámetros que inician una operación de creación, actualización o eliminación.

7. En la página Revisar *nombre de pila* confirme que se están importando los recursos correctos y, a continuación, elija Importar recursos. Esto ejecuta automáticamente el conjunto de cambios creado en el último paso.

Aparece el panel Eventos de la página Detalles de la pila para la nueva pila.

ImportTestStack

Delete

Update

Stack actions

Create stack

Stack info

Events

Resources

Outputs

Parameters

Template

Change sets

Events

Search events

Timestamp	Logical ID	Status	Status reason
2019-10-24 15:52:49 UTC-0700	ImportTestStack	IMPORT_COMPLETE	-
2019-10-24 15:52:47 UTC-0700	SampleIamUser	UPDATE_COMPLETE	-
2019-10-24 15:52:47 UTC-0700	SampleIamUser	UPDATE_IN_PROGRESS	Apply stack-level tags to imported resource if applicable.
2019-10-24 15:52:46 UTC-0700	SampleIamUser	IMPORT_COMPLETE	Resource import completed.
2019-10-24 15:52:45 UTC-0700	SampleIamUser	IMPORT_IN_PROGRESS	Resource import started.
2019-10-24 15:52:43 UTC-0700	ImportTestStack	IMPORT_IN_PROGRESS	User Initiated
2019-10-24 15:52:01 UTC-0700	ImportTestStack	REVIEW_IN_PROGRESS	User Initiated

8. (Opcional) Ejecute la detección de desviaciones en la pila para asegurarse de que la plantilla y la configuración real de los recursos importados coinciden. Para obtener más información sobre la detección de desviaciones, consulte [Detección de desviaciones en una pila de CloudFormation completa](#).
9. (Opcional) Si los recursos importados no coinciden con las configuraciones de plantilla esperadas, corrija las configuraciones de plantilla o actualice los recursos directamente. En este

tutorial, corregimos las configuraciones de plantilla para que coincidan con sus configuraciones reales.

- a. [Revierta la operación de importación](#) de los recursos afectados.
- b. Agregue de nuevo los destinos de importación a la plantilla, asegurándose de que las configuraciones de plantilla coincidan con las configuraciones reales.
- c. Repita los pasos del 2 al 8 con la plantilla modificada para importar de nuevo los recursos.

Crear una pila a partir de recursos existentes mediante la AWS CLI

1. Para saber qué propiedades identifican cada tipo de recurso en la plantilla, ejecute el comando `get-template-summary` y especifique la URL de S3 de la plantilla. Por ejemplo, el recurso `AWS::DynamoDB::Table` se puede identificar mediante la propiedad `TableName`. Para el recurso `GameTable` del ejemplo de plantilla, el valor de `TableName` es `Games`. Necesitará esta información en el siguiente paso.

```
aws cloudformation get-template-summary \
  --template-url https://amzn-s3-demo-bucket.s3.us-west-2.amazonaws.com/
  TemplateToImport.json
```

Para obtener más información, consulte [Identificadores de recursos](#).

2. Redacte una lista de los recursos reales de su plantilla y sus identificadores únicos en el siguiente formato de cadena JSON.

```
[{"ResourceType":"AWS::DynamoDB::Table","LogicalResourceId":"GameTable","ResourceIdentifier":
{"TableName":"Games"}},
{"ResourceType":"AWS::DynamoDB::Table","LogicalResourceId":"ServiceTable","ResourceIdentifier":
{"TableName":"Service"}}]
```

Si lo desea, también puede especificar los parámetros con formato JSON en un archivo de configuración.

Por ejemplo, para importar `ServiceTable` y `GameTable`, puede crear un archivo *ResourcesToImport.txt* que contenga la siguiente configuración.

```
[
  {
    "ResourceType":"AWS::DynamoDB::Table",
```

```

    "LogicalResourceId": "GamesTable",
    "ResourceIdentifier": {
        "TableName": "Games"
    }
},
{
    "ResourceType": "AWS::DynamoDB::Table",
    "LogicalResourceId": "ServiceTable",
    "ResourceIdentifier": {
        "TableName": "Service"
    }
}
]

```

3. Para crear un conjunto de cambios, utilice el siguiente comando `create-change-set` y sustituya el texto del marcador de posición. Para la opción `--change-set-type`, especifique un valor de **IMPORT**. Para la opción `--resources-to-import`, reemplace la cadena JSON de muestra por la cadena JSON real que acaba de crear.

```

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` no es compatible con YAML insertado. Los requisitos para escapar las comillas en la cadena JSON varían dependiendo de su terminal. Para obtener más información, consulte [Using quotation marks inside strings](#) en la Guía del usuario de la AWS Command Line Interface.

Como alternativa, también puede usar la URL de un archivo como entrada para la opción `--resources-to-import`, tal como se muestra en el siguiente ejemplo.

```
--resources-to-import file:///ResourcesToImport.txt
```

4. Revise el conjunto de cambios para asegurarse de que se importarán los recursos correctos.

```
aws cloudformation describe-change-set \  
  --change-set-name ImportChangeSet --stack-name TargetStack
```

5. Para iniciar el conjunto de cambios e importar los recursos, utilice el siguiente comando `execute-change-set` y sustituya el texto del marcador de posición. Una vez finalizada correctamente la operación (`IMPORT_COMPLETE`), los recursos se importan correctamente.

```
aws cloudformation execute-change-set \  
  --change-set-name ImportChangeSet --stack-name TargetStack
```

6. (Opcional) Ejecute la detección de desviaciones en la pila `IMPORT_COMPLETE` para asegurarse de que la plantilla y la configuración real de los recursos importados coinciden. Para obtener más información acerca de la detección de desviaciones, consulte [Detección de desviaciones en recursos de la pila individuales](#).

- a. Ejecute la detección de desviaciones en la pila especificada.

```
aws cloudformation detect-stack-drift --stack-name TargetStack
```

Si se ejecuta correctamente, este comando devuelve la siguiente salida de muestra.

```
{ "Stack-Drift-Detection-Id" : "624af370-311a-11e8-b6b7-500cexample" }
```

- b. Vea el progreso de una operación de detección de desviaciones para el ID de detección de desviaciones de pila especificado.

```
aws cloudformation describe-stack-drift-detection-status \  
  --stack-drift-detection-id 624af370-311a-11e8-b6b7-500cexample
```

- c. Vea la información de desviaciones de los recursos en los que se han comprobado las desviaciones de la pila especificada.

```
aws cloudformation describe-stack-resource-drifts --stack-name TargetStack
```

7. (Opcional) Si los recursos importados no coinciden con las configuraciones de plantilla esperadas, corrija las configuraciones de plantilla o actualice los recursos directamente. En este tutorial, corregimos las configuraciones de plantilla para que coincidan con sus configuraciones reales.
 - a. [Revierta la operación de importación](#) de los recursos afectados.
 - b. Agregue de nuevo los destinos de importación a la plantilla, asegurándose de que las configuraciones de plantilla coincidan con las configuraciones reales.
 - c. Repita los pasos del 3 al 6 con la plantilla modificada para importar de nuevo los recursos.

Importación de recursos existentes en una pila

En este tema se muestra cómo importar los recursos de AWS existentes a una pila existente al describirlos en una plantilla. Para buscar en su lugar los recursos existentes y generar automáticamente una plantilla que se pueda utilizar para importar los recursos existentes a CloudFormation o replicar los recursos en una cuenta nueva, consulte [Generar plantillas de los recursos existentes con el generador IaC](#).

Requisitos previos

Antes de comenzar, debe disponer de lo siguiente:

- Una plantilla que describa toda la pila, incluidos los recursos que ya forman parte de la pila y los recursos que se van a importar. Guarde la plantilla de manera local o en un bucket de Amazon S3.

Obtención de una copia de la plantilla de una pila en ejecución

1. Abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation/>.
 2. En la lista de pilas, seleccione la pila de la que desea recuperar la plantilla.
 3. En el panel de detalles de la pila, seleccione la pestaña Plantilla y, a continuación, seleccione Copiar en el portapapeles.
 4. Pegue el código en un editor de texto para empezar a agregar otros recursos a la plantilla.
- Para cada recurso que se desee importar, incluya lo siguiente:
 - Las propiedades y los valores de las propiedades que definen la configuración actual del recurso.
 - El identificador único por cada recurso; por ejemplo, por cada nombre. Para obtener más información, consulte [Identificadores de recursos](#).

- El [atributo DeletionPolicy](#).

Temas

- [Plantilla de ejemplo](#)
- [Importar un recurso existente a una pila mediante la Consola de administración de AWS](#)
- [Importar un recurso existente a una pila mediante la AWS CLI](#)

Plantilla de ejemplo

En este tutorial, se supone que utiliza la siguiente plantilla de ejemplo, llamada `TemplateToImport.json`, la cual especifica dos tablas de DynamoDB. `ServiceTable` actualmente forma parte de la pila y `GameTable` es la tabla que desea importar.

Note

Esta plantilla es solo un ejemplo. Para utilizarla en las pruebas propias, sustituya los recursos de ejemplo con los recursos de su cuenta.

```
{
  "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
            }
        }
    }
}

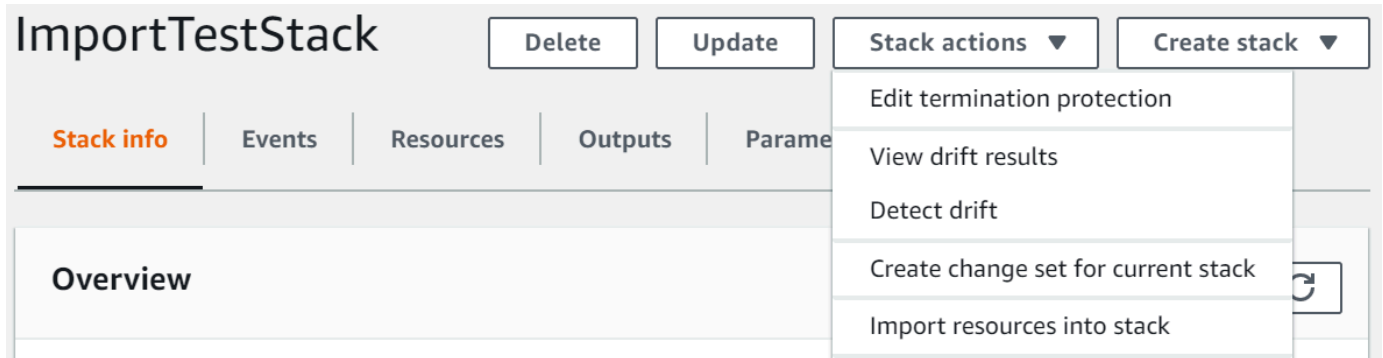
```

Importar un recurso existente a una pila mediante la Consola de administración de AWS

Note

La consola de CloudFormation no admite el uso de la función intrínseca `Fn::Transform` al importar recursos. Puede usar la AWS CLI para importar recursos que usen la función intrínseca `Fn::Transform`.

1. Inicie sesión en la Consola de administración de AWS y abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation>.
2. En la página Pilas, elija la pila a la que desea importar recursos.
3. Elija Acciones de pila, y, a continuación, elija Importar recursos en la pila.



4. Revise la página Información general de importación y, a continuación, elija Siguiente.
5. En la página Especificar plantilla proporcione la plantilla actualizada utilizando uno de los métodos siguientes y, a continuación, elija Siguiente.
 - Seleccione URL de Amazon S3, y, a continuación, especifique la URL de la plantilla en el cuadro de texto.
 - Seleccione Cargar un archivo de plantilla y, a continuación, busque la plantilla.
6. En la página Identificar recursos, identifique cada recurso de destino. Para obtener más información, consulte [Identificadores de recursos](#).
 - a. En Propiedad del identificador, elija el tipo de identificador de recurso. Por ejemplo, el recurso `AWS::DynamoDB::Table` se puede identificar mediante la propiedad `TableName`.
 - b. En Valor del identificador, escriba el valor de la propiedad real. Por ejemplo, el `TableName` para el recurso `GameTable` en el ejemplo de plantilla es *Games*.
 - c. Elija Siguiente.
7. En la página Especificar detalles de pila actualice los parámetros y, a continuación, elija Siguiente. Esto crea automáticamente un conjunto de cambios.

 Note

La operación de importación produce un error si modifica parámetros que inician una operación de creación, actualización o eliminación.

- En la página Revisar **nombre de pila** revise los recursos que desea importar y, a continuación, elija Importar recursos. Esto ejecuta automáticamente el conjunto de cambios creado en el último paso. En este momento, se aplican todas las etiquetas de nivel de pila a los recursos importados. Para obtener más información, consulte [Configurar las opciones la pila](#).

Aparece la página Eventos de la pila.

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

- (Opcional) Ejecute la detección de desviaciones en la pila para asegurarse de que la plantilla y la configuración real de los recursos importados coinciden. Para obtener más información sobre la detección de desviaciones, consulte [Detección de desviaciones en una pila de CloudFormation completa](#).
- (Opcional) Si los recursos importados no coinciden con las configuraciones de plantilla esperadas, corrija las configuraciones de plantilla o actualice los recursos directamente. Para obtener más información acerca de la importación de recursos desviados, consulte [Resolver desviaciones con una operación de importación](#).

Importar un recurso existente a una pila mediante la AWS CLI

- Para saber qué propiedades identifican cada tipo de recurso en la plantilla, ejecute el comando `get-template-summary` y especifique la URL de S3 de la plantilla. Por ejemplo, el recurso `AWS::DynamoDB::Table` se puede identificar mediante la propiedad `TableName`. Para el recurso `GameTable` del ejemplo de plantilla, el valor de `TableName` es `Games`. Necesitará esta información en el siguiente paso.

```
aws cloudformation get-template-summary \
```

```
--template-url https://amzn-s3-demo-bucket.s3.us-west-2.amazonaws.com/
TemplateToImport.json
```

Para obtener más información, consulte [Identificadores de recursos](#).

2. Redacte una lista de recursos reales para importar y sus identificadores únicos en el siguiente formato de cadena JSON.

```
[{"ResourceType":"AWS::DynamoDB::Table","LogicalResourceId":"GamesTable","ResourceIdentifier":{
{"TableName":"Games"}}]
```

Si lo desea, también puede especificar los parámetros con formato JSON en un archivo de configuración.

Por ejemplo, para importar GamesTable, puede crear un archivo *ResourcesToImport.txt* que contenga la siguiente configuración.

```
[
{
  "ResourceType":"AWS::DynamoDB::Table",
  "LogicalResourceId":"GamesTable",
  "ResourceIdentifier": {
    "TableName":"Games"
  }
}
]
```

3. Para crear un conjunto de cambios, utilice el siguiente comando create-change-set y sustituya el texto del marcador de posición. Para la opción --change-set-type, especifique un valor de **IMPORT**. Para la opción --resources-to-import, reemplace la cadena JSON de muestra por la cadena JSON real que acaba de crear.

```
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` no es compatible con YAML insertado. Los requisitos para escapar las comillas en la cadena JSON varían dependiendo de su terminal. Para obtener más información, consulte [Using quotation marks inside strings](#) en la Guía del usuario de la AWS Command Line Interface.

Como alternativa, también puede usar la URL de un archivo como entrada para la opción `--resources-to-import`, tal como se muestra en el siguiente ejemplo.

```
--resources-to-import file:///ResourcesToImport.txt
```

4. Revise el conjunto de cambios para asegurarse de que se importarán los recursos correctos.

```
aws cloudformation describe-change-set \  
  --change-set-name ImportChangeSet --stack-name TargetStack
```

5. Para iniciar el conjunto de cambios e importar los recursos, utilice el siguiente comando `execute-change-set` y sustituya el texto del marcador de posición. En este momento, se aplican todas las etiquetas de nivel de pila a los recursos importados. Para obtener más información, consulte [Configurar las opciones la pila](#). Una vez finalizada correctamente la operación (`IMPORT_COMPLETE`), los recursos se importan correctamente.

```
aws cloudformation execute-change-set \  
  --change-set-name ImportChangeSet --stack-name TargetStack
```

6. (Opcional) Ejecute la detección de desviaciones en la pila `IMPORT_COMPLETE` para asegurarse de que la plantilla y la configuración real de los recursos importados coinciden. Para obtener más información sobre la detección de desviaciones, consulte [Detección de desviaciones en una pila de CloudFormation completa](#).

- a. Ejecute la detección de desviaciones en la pila especificada.

```
aws cloudformation detect-stack-drift --stack-name TargetStack
```

Si se ejecuta correctamente, este comando devuelve la siguiente salida de muestra.

```
{ "Stack-Drift-Detection-Id" : "624af370-311a-11e8-b6b7-500cexample" }
```

- b. Vea el progreso de una operación de detección de desviaciones para el ID de detección de desviaciones de pila especificado.

```
aws cloudformation describe-stack-drift-detection-status \  
  --stack-drift-detection-id 624af370-311a-11e8-b6b7-500cexample
```

- c. Vea la información de desviaciones de los recursos en los que se han comprobado las desviaciones de la pila especificada.

```
aws cloudformation describe-stack-resource-drifts --stack-name TargetStack
```

7. (Opcional) Si los recursos importados no coinciden con las configuraciones de plantilla esperadas, corrija las configuraciones de plantilla o actualice los recursos directamente. Para obtener más información acerca de la importación de recursos desviados, consulte [Resolver desviaciones con una operación de importación](#).

Mover recursos entre pilas

Con la característica `resource import`, puede mover recursos entre pilas o refactorizar pilas. En primer lugar, debe agregar una política de eliminación `Retain` al recurso que desea mover para asegurarse de que el recurso se conserva al eliminarlo de la pila de origen e importarlo a la pila de destino.

Si es nuevo en la importación, le recomendamos que primero revise la información introductoria del tema [Cómo importar recursos de AWS a una pila de CloudFormation](#).

Important

No todos los recursos admiten operaciones de importación. Consulte [Recursos que admiten operaciones de importación](#) antes de eliminar un recurso de la pila. Si elimina un recurso que no admite operaciones de importación de su pila, no puede importar el recurso a otra pila ni devolverlo a la pila de origen.

Refactorización de una pila mediante la Consola de administración de AWS

1. En la plantilla de origen, especifique una Retain [DeletionPolicy](#) para el recurso que desea mover.

En la siguiente plantilla de origen de ejemplo, Games es el destino de esta refactorización.

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. Abra la consola de CloudFormation para realizar una actualización de la pila y aplicar la política de eliminación.
 - a. En la página Stacks (Pilas) con la pila seleccionada, elija Update (Actualizar).
 - b. En Preparar plantilla, elija Reemplazar la plantilla actual.
 - c. En Specify template (Especificar plantilla), proporcione la plantilla de origen actualizada con el atributo DeletionPolicy en GamesTable y, a continuación, elija Next (Siguiente).
 - Elija Amazon S3 URL (URL de Amazon S3) y, a continuación, especifique la URL de la plantilla de origen actualizada en el cuadro de texto.
 - Elija Upload a template file (Cargar un archivo de plantilla) y, a continuación, busque el archivo de la plantilla de origen actualizada.
 - d. En la página Specify stack details (Especificar detalles de pila) no se requieren cambios. Elija Next (Siguiente).
 - e. La página Configure stack options (Configurar opciones de pila), no precisa cambios. Elija Siguiente.
 - f. En la página Revisar **SourceStackName**, revise los cambios. Si la plantilla contiene recursos de IAM, seleccione I acknowledge that this template may create IAM resources (Confirmando que esta plantilla pueda crear recursos de IAM) para especificar que desea utilizar recursos de IAM en la plantilla. Para obtener más información sobre recursos de IAM en plantillas, consulte [Control del acceso a CloudFormation con AWS Identity and](#)

[Access Management](#). A continuación, actualice la pila de origen mediante la creación de un conjunto de cambios o actualice la pila de origen directamente.

3. Elimine el recurso, los parámetros relacionados y las salidas de la plantilla de origen y, a continuación, agréguelos a la plantilla de destino.

La plantilla de origen ahora tiene el siguiente aspecto.

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
        }
      }
    }
  }
}
```

La siguiente plantilla de destino de ejemplo tiene actualmente el recurso `PlayersTable` y ahora también contiene `GamesTable`.

Example JSON

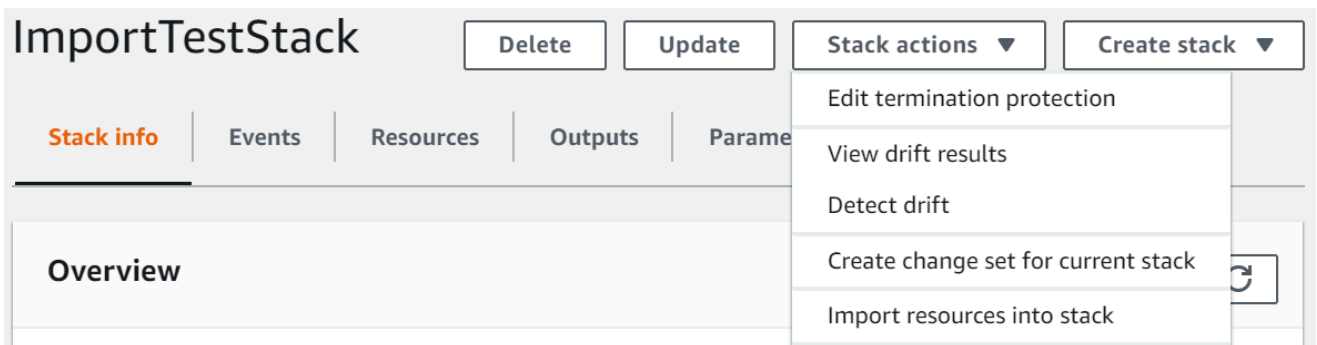
```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "Import test",
  "Resources": {
    "PlayersTable": {
      "Type": "AWS::DynamoDB::Table",
      "Properties": {
        "TableName": "Players",
        "AttributeDefinitions": [
          {
            "AttributeName": "key",
            "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. Repita los pasos 2 y 3 para actualizar la pila de origen de nuevo, esta vez con el fin de eliminar el recurso de destino de la pila.
5. Realice una operación de importación para agregar GameSTable a la pila de destino.
 - a. En la página Stacks (Pilas), con la pila principal seleccionada, elija Stack actions (Acciones de pila) y, a continuación, elija Import resources into stack (Importar recursos a pila).



- b. Lea la página de información general sobre la importación para obtener una lista de los elementos que debe proporcionar durante esta operación. A continuación, elija Siguiente.
- c. En la página Specify template (Especificar plantilla) haga una de las siguientes acciones y, a continuación, elija Next (Siguiente).
 - Elija Amazon S3 URL (URL de Amazon S3) y, a continuación, especifique la URL en el cuadro de texto.
 - Elija Upload a template file (Cargar un archivo de plantilla) y, a continuación, busque el archivo que se va a cargar.
- d. En la página Identify resources (Identificar recursos), identifique el recurso que va a mover (en este ejemplo, GameSTable). Para obtener más información, consulte [Identificadores de recursos](#).

1. En Propiedad del identificador, elija el tipo de identificador de recurso. Por ejemplo, un recurso `AWS::DynamoDB::Table` se puede identificar mediante la propiedad `TableName`.

2. En Valor del identificador, escriba el valor de la propiedad real. Por ejemplo, `GamesTables`.

3. Elija Siguiente.

e. En la página Especificar detalles de pila modifique los parámetros y, a continuación, elija Siguiente. Esto crea automáticamente un conjunto de cambios.



Important

La operación de importación produce un error si modifica parámetros que inician una operación de creación, actualización o eliminación.

f. En la página Revisar `TargetStackName`, confirme que se está importando el recurso correcto y, a continuación, elija Importar recursos. Esto inicia automáticamente el conjunto de cambios creado en el último paso. En este momento, se aplican todas las [etiquetas de nivel de pila](#) a los recursos importados.

g. Se muestra el panel Events (Eventos) de la página Stack details (Detalles de la pila) de la pila principal.
- ImportTestStack

DeleteUpdateStack actions▼Create stack▼

Stack infoEventsResourcesOutputsParametersTemplateChange 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
- Cómo importar recursos de AWS de forma manual

Versión de API 2010-05-15 855

 Note

No es necesario ejecutar la detección de desviación en la pila principal después de esta operación de importación porque el recurso `AWS::CloudFormation::Stack` ya está administrado por CloudFormation.

Refactorización de una pila mediante la AWS CLI

1. En la plantilla de origen, especifique una `Retain` [DeletionPolicy](#) para el recurso que desea mover.

En la siguiente plantilla de origen de ejemplo, `GameTable` es el destino de esta refactorización.

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. Actualice la pila de origen para aplicar la política de eliminación al recurso.

```
aws cloudformation update-stack --stack-name SourceStackName
```

3. Elimine el recurso, los parámetros relacionados y las salidas de la plantilla de origen y, a continuación, agréguelos a la plantilla de destino.

La plantilla de origen ahora tiene el siguiente aspecto.

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
            }
        }
    }
}

```

La siguiente plantilla de destino de ejemplo tiene actualmente el recurso `PlayersTable` y ahora también contiene `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. Actualice la pila de origen para eliminar el recurso (GamesTable) y sus parámetros y salidas relacionados de la pila.

```
aws cloudformation update-stack --stack-name SourceStackName
```

5. Redacte una lista de recursos reales para importar y sus identificadores únicos en el siguiente formato de cadena JSON. Para obtener más información, consulte [Identificadores de recursos](#).

```
[{"ResourceType":"AWS::DynamoDB::Table","LogicalResourceId":"GamesTable","ResourceIdentifier":{"TableName":"Games"}}]
```

Si lo desea, también puede especificar los parámetros con formato JSON en un archivo de configuración.

Por ejemplo, para importar GamesTable, puede crear un archivo *ResourcesToImport.txt* que contenga la siguiente configuración.

```
[
  {
    "ResourceType":"AWS::DynamoDB::Table",
    "LogicalResourceId":"GamesTable",
    "ResourceIdentifier": {
      "TableName":"Games"
    }
  }
]
```

6. Para crear un conjunto de cambios, utilice el siguiente comando `create-change-set` y sustituya el texto del marcador de posición. Para la opción `--change-set-type`, especifique un valor de **IMPORT**. Para la opción `--resources-to-import`, reemplace la cadena JSON de muestra por la cadena JSON real que acaba de crear.

```
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` no es compatible con YAML insertado. Los requisitos para escapar las comillas en la cadena JSON varían dependiendo de su terminal. Para

obtener más información, consulte [Using quotation marks inside strings](#) en la Guía del usuario de la AWS Command Line Interface.

Como alternativa, también puede usar la URL de un archivo como entrada para la opción `--resources-to-import`, tal como se muestra en el siguiente ejemplo.

```
--resources-to-import file:///ResourcesToImport.txt
```

7. Revise el conjunto de cambios para asegurarse de que se está importando el recurso correcto a la pila de destino.

```
aws cloudformation describe-change-set \  
  --change-set-name ImportChangeSet
```

8. Para iniciar el conjunto de cambios e importar el recurso, utilice el siguiente comando `execute-change-set` y sustituya el texto del marcador de posición. En este momento, se aplican todas las etiquetas de nivel de pila a los recursos importados. Una vez finalizada debidamente la operación (`IMPORT_COMPLETE`), el recurso se importa correctamente.

```
aws cloudformation execute-change-set \  
  --change-set-name ImportChangeSet --stack-name TargetStackName
```

Note

No es necesario ejecutar la detección de desviación en la pila de destino después de esta operación de importación porque el recurso ya está administrado por CloudFormation.

Anidamiento de una pila existente

Utilice la función `resource import` para anidar una pila existente dentro de otra pila existente. Las pilas anidadas son componentes comunes a los que se declara y hace referencia desde otras plantillas. De esta forma, puede evitar copiar y pegar las mismas configuraciones en sus plantillas y simplificar las actualizaciones de pilas. Si tiene una plantilla para un componente común, puede utilizar el recurso `AWS::CloudFormation::Stack` para hacer referencia a esta plantilla desde otra

plantilla. Para obtener más información sobre las pilas anidadas, consulte [División de una plantilla en piezas reutilizables utilizando pilas anidadas](#).

CloudFormation solo admite un nivel de uso de anidamiento utilizando `resource import`. Esto significa que no puede importar una pila a una pila secundaria ni importar una pila que tenga secundarias.

Si es nuevo en la importación, le recomendamos que primero revise la información introductoria del tema [Cómo importar recursos de AWS a una pila de CloudFormation de forma manual](#).

Validación de importación de pila anidada

Durante una operación de importación de pila anidada, CloudFormation realiza las siguientes validaciones.

- La definición anidada `AWS::CloudFormation::Stack` en la plantilla de pila principal coincide con la plantilla de la pila anidada real.
- Las etiquetas de la definición anidada `AWS::CloudFormation::Stack` en la plantilla de pila principal coinciden con las etiquetas del recurso de pila anidado real.

Anidar una pila existente mediante la Consola de administración de AWS

1. Agregue el recurso `AWS::CloudFormation::Stack` a la plantilla de pila principal con una `Retain` [DeletionPolicy](#). En el siguiente plantilla de pila principal de ejemplo, `MyNestedStack` es el destino de la importación.

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

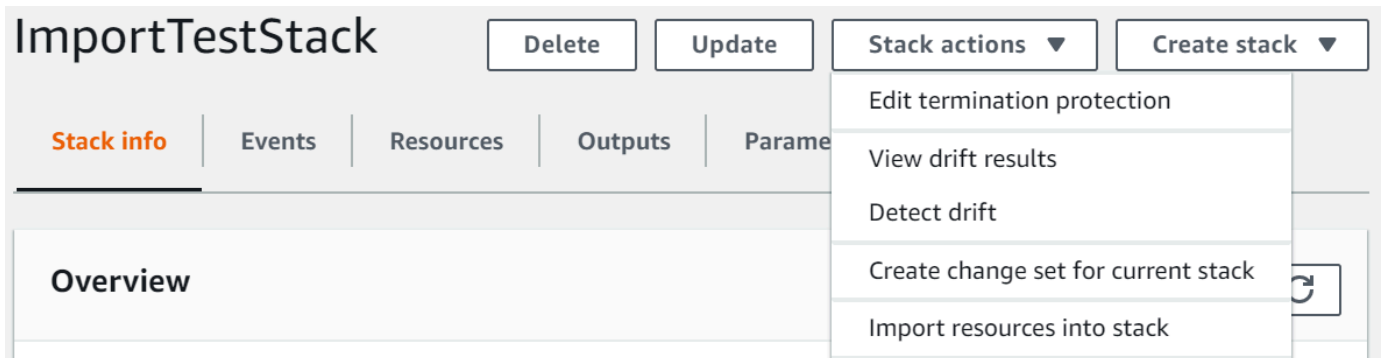
```

```

    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. Abra la consola de CloudFormation.
3. En la página Stacks (Pilas), con la pila principal seleccionada, elija Stack actions (Acciones de pila) y, a continuación, elija Import resources into stack (Importar recursos a pila).



4. Lea la página de información general sobre la importación para obtener una lista de los elementos que debe proporcionar durante esta operación. A continuación, elija Siguiente.
5. En la página Specify template (Especificar plantilla), proporcione la plantilla principal actualizada mediante uno de los métodos siguientes y, a continuación, elija Next (Siguiente).
 - Seleccione URL de Amazon S3, y, a continuación, especifique la URL de la plantilla en el cuadro de texto.
 - Seleccione Cargar un archivo de plantilla y, a continuación, busque la plantilla.
6. En la página Identify resources (Identificar recursos), identifique el recurso `AWS::CloudFormation::Stack`.
 - a. En Propiedad del identificador, elija el tipo de identificador de recurso. Por ejemplo, un recurso `AWS::CloudFormation::Stack` se puede identificar mediante la propiedad `StackId`.

- b. En Valor del identificador, escriba el ARN de la pila que va a importar. Por ejemplo, `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. Elija Siguiente.
7. En la página Especificar detalles de pila modifique los parámetros y, a continuación, elija Siguiente. Esto crea automáticamente un conjunto de cambios.
-  **Important**
La operación de importación produce un error si modifica parámetros que inician una operación de creación, actualización o eliminación.
8. En la página Revisar **MyParentStack**, confirme que se está importando el recurso correcto y, a continuación, elija Importar recursos. Esto ejecuta automáticamente el conjunto de cambios creado en el último paso. En este momento, se aplican todas las etiquetas de nivel de pila a los recursos importados.
9. Se muestra el panel Events (Eventos) de la página Stack details (Detalles de la pila) de la pila principal.

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

No es necesario ejecutar la detección de desviación en la pila principal después de esta operación de importación porque el recurso `AWS::CloudFormation::Stack` ya estaba administrado por CloudFormation.

Anidar una pila existente mediante la AWS CLI

1. Agregue el recurso `AWS::CloudFormation::Stack` a la plantilla de pila principal con una `Retain` [DeletionPolicy](#). En el siguiente ejemplo, la plantilla principal `MyNestedStack` es el destino de la importación.

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. Componga una cadena JSON tal como se muestra en el siguiente ejemplo, con estas modificaciones:

- Sustituya *MyNestedStack* por el ID lógico del recurso de destino tal como se especifica en la plantilla.
- Sustituya *arn:aws:cloudformation:us-west-2:12345678910:stack/mystack/5b918d10-cd98-11ea-90d5-0a9cd3354c10* por el ARN de la pila que desee importar.

```

[{"ResourceType": "AWS::CloudFormation::Stack", "LogicalResourceId": "MyNestedStack", "Resource
{"StackId": "arn:aws:cloudformation:us-east-2:123456789012:stack/mystack/5b918d10-
cd98-11ea-90d5-0a9cd3354c10"}}]

```

Si lo desea, también puede especificar los parámetros en un archivo de configuración.

Por ejemplo, para importar *MyNestedStack*, puede crear un archivo *ResourcesToImport.txt* que contenga la siguiente configuración.

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. Para crear un conjunto de cambios, utilice el siguiente comando `create-change-set` y sustituya el texto del marcador de posición. Para la opción `--change-set-type`, especifique un valor de **IMPORT**. Para la opción `--resources-to-import`, reemplace la cadena JSON de muestra por la cadena JSON real que acaba de crear.

```
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` no es compatible con YAML insertado. Los requisitos para escapar las comillas en la cadena JSON varían dependiendo de su terminal. Para obtener más información, consulte [Using quotation marks inside strings](#) en la Guía del usuario de la AWS Command Line Interface.

Como alternativa, también puede usar la URL de un archivo como entrada para la opción `--resources-to-import`, tal como se muestra en el siguiente ejemplo.

```
--resources-to-import file://ResourcesToImport.txt
```

Si se ejecuta correctamente, este comando devuelve la siguiente salida de muestra.

```
{
  "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. Revise el conjunto de cambios para asegurarse de que se está importando la pila correcta.

```
aws cloudformation describe-change-set --change-set-name ImportChangeSet
```

5. Para iniciar el conjunto de cambios e importar la pila a la pila principal de origen, utilice el siguiente comando `execute-change-set` y sustituya el texto del marcador de posición. En este momento, se aplican todas las [etiquetas de nivel de pila](#) a los recursos importados. Una vez completada correctamente la operación de importación (`IMPORT_COMPLETE`), la pila se anidará correctamente.

```
aws cloudformation execute-change-set --change-set-name ImportChangeSet
```

Note

No es necesario ejecutar la detección de desviación en la pila principal después de esta operación de importación porque el recurso `AWS::CloudFormation::Stack` ya está administrado por CloudFormation.

Cómo importar recursos de AWS a una pila de CloudFormation automáticamente

Ahora puede importar automáticamente recursos con nombre al crear o actualizar pilas de CloudFormation. Un recurso con nombre es aquel al que se le ha asignado un nombre personalizado. Para obtener más información, consulte [Tipo de nombre](#) en la Referencia de plantillas de CloudFormation.

Cuando inicia una importación automática, CloudFormation verifica si existen recursos que coincidan con su plantilla y los importa durante la implementación. Para pilas anidadas, cree el conjunto de cambios desde la pila raíz.

Una vez finalizada la importación y antes de realizar operaciones de pila posteriores, se recomienda ejecutar la detección de desviaciones en los recursos importados. La detección de desviaciones garantiza que la configuración de la plantilla coincida con la configuración real. Para obtener más información, consulte [Detección de desviaciones en una pila de CloudFormation completa](#).

Para importar un recurso, este debe cumplir con los siguientes requisitos:

- El recurso debe tener un nombre personalizado estático definido en la plantilla. Los nombres dinámicos (que usan `!Ref` u otras funciones) no son compatibles por el momento.
- El recurso debe tener una `DeletionPolicy` de `Retain` o `RetainExceptOnCreate`.
- El recurso no debe pertenecer a otra pila de CloudFormation.
- El tipo de recurso debe admitir operaciones de importación de CloudFormation. Para obtener más información, consulte [Tipo de recurso compatible](#).
- El identificador principal del tipo de recurso debe estar en la plantilla. No se admiten los identificadores principales con propiedades de solo lectura. Para saber cuál es el identificador principal de un tipo, busque la propiedad `primaryIdentifier` en el esquema de recursos. Para obtener más información sobre la propiedad, consulte [primaryIdentifier](#).

Example Ejemplo de importación automática

El siguiente ejemplo usa un conjunto de cambios, `CreateChangeSet`, para crear una pila llamada `my-stack` basada en un archivo de plantilla, `template.yaml`, e importa automáticamente los recursos que coincidan.

```
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
```

Solución de problemas

Cuando se produce un error en la importación automática, puede tomar las siguientes medidas:

- Verifique que el nombre del recurso en la plantilla coincida exactamente con el nombre del recurso
- Verifique que el recurso no sea administrado por otra pila
- Asegúrese de que el tipo de recurso admita operaciones de importación
- Verifique que la plantilla incluya todas las propiedades requeridas para el tipo de recurso

Reversión de una operación de importación

Para revertir una operación de importación, especifique una política de eliminación `Retain` para el recurso que desea eliminar de la plantilla a fin de asegurarse de que se conserva al eliminarlo de la pila.

Revertir una operación de importación mediante la Consola de administración de AWS

1. Especifique una `Retain` [DeletionPolicy](#) para los recursos que desea eliminar de la pila. En la plantilla de ejemplo siguiente, `GameTable` es el destino de esta operación de reversión.

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. Abra la consola de CloudFormation para realizar una actualización de la pila y aplicar la política de eliminación.
 - a. En la página Stacks (Pilas), con la pila seleccionada, elija Update (Actualizar) y, a continuación, elija Update stack (standard) (Actualizar pila [estándar]).
 - b. En Preparar plantilla, elija Reemplazar la plantilla actual.
 - c. En Specify template (Especificar plantilla), proporcione la plantilla de origen actualizada con el atributo DeletionPolicy en GamesTable y, a continuación, elija Next (Siguiente).
 - Elija Amazon S3 URL (URL de Amazon S3) y, a continuación, especifique la URL de la plantilla de origen actualizada en el cuadro de texto.

- Elija Upload a template file (Cargar un archivo de plantilla) y, a continuación, busque el archivo de la plantilla de origen actualizada.
- d. En la página Specify stack details (Especificar detalles de pila) no se requieren cambios. Elija Next (Siguiente).
 - e. La página Configure stack options (Configurar opciones de pila), no precisa cambios. Elija Siguiente.
 - f. En la página Revisar **MyStack**, revise los cambios. Si la plantilla contiene recursos de IAM, seleccione I acknowledge that this template may create IAM resources (Confirmando que esta plantilla pueda crear recursos de IAM) para especificar que desea utilizar recursos de IAM en la plantilla. Para obtener más información, consulte [Reconocimiento de recursos de IAM en plantillas de CloudFormation](#). A continuación, actualice la pila de origen mediante la creación de un conjunto de cambios o actualice la pila de origen directamente.
3. Elimine el recurso, los parámetros relacionados y las salidas de la plantilla de la pila. En este ejemplo, la plantilla ahora tiene el siguiente aspecto.

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. Repita el paso 2 para eliminar el recurso (GamesTable) y sus parámetros y salidas relacionados de la pila.

Revertir una operación de importación mediante la AWS CLI

1. Especifique una Retain [DeletionPolicy](#) para los recursos que desea eliminar de la pila. En la plantilla de ejemplo siguiente, GamesTable es el destino de esta operación de reversión.

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. Actualice la pila para aplicar la política de eliminación al recurso.

```
aws cloudformation update-stack --stack-name MyStack
```

3. Elimine el recurso, los parámetros relacionados y las salidas de la plantilla de la pila. En este ejemplo, la plantilla ahora tiene el siguiente aspecto.

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. Actualice la pila para eliminar el recurso (GameTable) y sus parámetros y salidas relacionados de la pila.

```
aws cloudformation update-stack --stack-name MyStack
```

Refactorización de pilas

Con la refactorización de pilas, puede reorganizar los recursos de las pilas de CloudFormation y, al mismo tiempo, conservar las propiedades y los datos de los recursos existentes. Puede mover recursos entre pilas, dividir pilas grandes en pilas más pequeñas o combinar varias pilas en una sola.

Temas

- [Cómo funciona la refactorización de pilas](#)
- [Consideraciones de la refactorización de pilas](#)
- [Requisitos previos](#)
- [Refactorización de las pilas \(consola\)](#)
- [Refactorización de las pilas \(AWS CLI\)](#)
- [Limitaciones de recursos](#)

Cómo funciona la refactorización de pilas

La refactorización de las pilas implica estas fases:

1. Evaluación de su infraestructura actual: revise sus pilas y recursos de CloudFormation existentes para identificar las oportunidades de refactorización de pilas.
2. Planificación de su refactorización: defina cómo deben organizarse los recursos. Tenga en cuenta las dependencias, las convenciones de nomenclatura y los límites operativos. Esto puede afectar a la validación de CloudFormation más adelante.
3. Determinación de las pilas de destino: decida en qué pilas refactorizará los recursos. Puede mover los recursos entre al menos 2 pilas (mediante la consola) y un máximo de 5 pilas (mediante la AWS CLI). Los recursos se pueden mover entre pilas anidadas.
4. Actualice las plantillas: modifique las plantillas de CloudFormation para que reflejen el cambio planificado, como mover las definiciones de recursos entre plantillas. Puede cambiar el nombre de los ID lógicos durante este proceso.
5. Cree la refactorización de la pila: proporcione una lista de los nombres y plantillas de la pila que desee refactorizar.
6. Revise el impacto de la refactorización y resuelva cualquier conflicto: CloudFormation valida las plantillas que proporciona y comprueba las dependencias entre pilas, los tipos de recursos con problemas de actualización de etiquetas y los conflictos de ID lógicos de los recursos.

Si la validación se realiza correctamente, CloudFormation generará una vista previa de las acciones de refactorización que se llevarán a cabo durante la ejecución.

Si la validación falla, resuelva los problemas identificados y vuelva a intentarlo. Si hay conflictos, proporcione una asignación de ID lógica del recurso que muestre el origen y el destino del recurso en conflicto.

7. Ejecute la refactorización: después de confirmar que los cambios se alinean con los objetivos de refactorización, complete la refactorización de la pila.
8. Supervise: realice un seguimiento del estado de ejecución para garantizar que la operación se complete correctamente.

Consideraciones de la refactorización de pilas

Al refactorizar las pilas, tenga en cuenta lo siguiente:

- La refactorización de pilas se limita a reorganizar los recursos existentes. No puede crear ni eliminar recursos, modificar las configuraciones de los recursos ni cambiar o agregar nuevos parámetros, condiciones o asignaciones durante la refactorización. Para realizar estos cambios, actualice primero las pilas y, a continuación, realice la refactorización de las pilas.
- No puede refactorizar el mismo recurso en varias pilas.
- No puede refactorizar los recursos que hacen referencia a pseudoparámetros cuyos valores difieren entre las pilas de origen y destino, por ejemplo, `AWS::StackName`.
- CloudFormation no admite pilas vacías. Si la refactorización deja una pila sin recursos, primero debe añadir al menos un recurso a esa pila antes de ejecutar [create-stack-refactor](#). Puede ser un recurso simple como `AWS::SNS::Topic` o `AWS::CloudFormation::WaitCondition`. Por ejemplo:

```
Resources:
  MySimpleSNSTopic:
    Type: AWS::SNS::Topic
    Properties:
      DisplayName: MySimpleTopic
```

- La refactorización de pilas no admite pilas que tengan políticas de apilamiento adjuntas, independientemente de lo que las políticas permitan o denieguen.

Requisitos previos

Para refactorizar las pilas, debe haber creado ya las plantillas revisadas.

Utilice el comando [get-template](#) para recuperar las plantillas de CloudFormation de las pilas que desee refactorizar.

```
aws cloudformation get-template --stack-name Stack1
```

Una vez que tenga la plantilla, utilice el entorno de desarrollo integrado (IDE) que prefiera para actualizarla y utilizar la estructura y la organización de recursos deseadas.

Refactorización de las pilas (consola)

Utilice el siguiente procedimiento para refactorizar una pila mediante la consola.

Refactorización de las pilas

1. Inicie sesión en la Consola de administración de AWS y abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation>.
2. En la barra de navegación de la parte superior de la pantalla, elija la Región de AWS en donde se encuentran las pilas.
3. En el panel de navegación de la izquierda, seleccione Refactorizaciones de la pila.
4. En la página de refactorizaciones de la pila, seleccione Iniciar la refactorización de la pila.
5. En Descripción, proporcione una descripción que lo ayude a identificar la refactorización de la pila. A continuación, elija Siguiente.
6. Para la pila 1, haga lo siguiente:

- a. Elija Actualizar la plantilla para una pila existente o Crear una pila nueva.

Si seleccionó Actualizar la plantilla para una pila existente, seleccione una pila existente de la lista. O elija Introducir un ARN de pila para introducir el ARN de una pila existente.

Si seleccionó Crear nueva pila, en Nombre de pila, proporcione un nombre para la nueva pila.

- b. En Reemplazar una plantilla existente por una plantilla refactorizada, seleccione URL de Amazon S3 o Cargar un archivo de plantilla para cargar la plantilla deseada para la pila 1.
- c. Elija Siguiente.

7. Para la pila 2, haga lo siguiente:

- a. Elija Actualizar la plantilla para una pila existente o Crear una pila nueva.

Si seleccionó Actualizar la plantilla para una pila existente, seleccione una pila existente de la lista. O elija Introducir un ARN de pila para introducir el ARN de una pila existente.

Si seleccionó Crear nueva pila, en Nombre de pila, proporcione un nombre para la nueva pila.

- b. En Reemplazar la plantilla existente por una plantilla refactorizada, seleccione URL de Amazon S3 o Cargar archivo de plantilla para cargar la plantilla deseada para la pila 2.
- c. Elija Siguiente.

8. En la página Especificar los cambios de nombre de los ID de los recursos lógicos, asegúrese de que CloudFormation sepa cómo refactorizar las pilas asignando los recursos que se muestran a sus ID lógicos correctos. Como parte de la refactorización de la pila, si se ha cambiado el ID

lógico de algún recurso, debe especificar el nombre de la pila de origen, el ID lógico original, el nombre de la pila de destino y el ID lógico al que se le ha cambiado el nombre. En algunos casos, es posible que la consola de CloudFormation detecte automáticamente la asignación de recursos y, antes de continuar, solo tiene que comprobar que la asignación de recursos rellenada previamente es correcta.

9. Elija Siguiente.
10. En la página Revisar y ejecutar, revise todas las selecciones de los pasos anteriores y confirme que todo está configurado correctamente.
11. Cuando tenga todo listo para refactorizar las pilas, elija Ejecutar la refactorización de la pila.

Refactorización de las pilas (AWS CLI)

Los comandos de la AWS CLI para la refactorización de pilas incluyen:

- [create-stack-refactor](#) para validar y generar una vista previa de los cambios planificados.
- [describe-stack-refactor](#) para recuperar el estado y los detalles de una operación de refactorización de pilas.
- [execute-stack-refactor](#) para completar la operación de refactorización de pilas validada.
- [list-stack-refactors](#) para enumerar todas las operaciones de refactorización de pilas de su cuenta con su estado actual e información básica.
- [list-stack-refactor-actions](#) para mostrar una vista previa de las acciones específicas que CloudFormation realizará en cada pila y recurso durante la ejecución de la refactorización.

Utilice el siguiente procedimiento para crear una pila mediante la AWS CLI.

Refactorización de las pilas

1. Utilice el comando [create-stack-refactor](#) y proporcione los nombres de las pilas y las plantillas actualizadas para la refactorización de pilas. Incluya la opción `--enable-stack-creation` para permitir que CloudFormation cree pilas nuevas si aún no existen.

```
aws cloudformation create-stack-refactor \
  --stack-definitions \
    StackName=Stack1,TemplateBody@=file://template1-updated.yaml \
    StackName=Stack2,TemplateBody@=file://template2-updated.yaml \
  --enable-stack-creation
```

El comando devuelve un `StackRefactorId` que utilizará en pasos posteriores.

```
{
  "StackRefactorId": "9c384f70-4e07-4ed7-a65d-fee5eb430841"
}
```

Si se detectan conflictos durante la validación de la plantilla (lo cual puede confirmar en el siguiente paso), utilice el comando [create-stack-refactor](#) con la opción `--resource-mappings`.

```
aws cloudformation create-stack-refactor \
  --stack-definitions \
    StackName=Stack1,TemplateBody@=file://template1-updated.yaml \
    StackName=Stack2,TemplateBody@=file://template2-updated.yaml \
  --enable-stack-creation \
  --resource-mappings file://resource-mapping.json
```

A continuación se muestra un ejemplo de un archivo `resource-mapping.json`.

```
[
  {
    "Source": {
      "StackName": "Stack1",
      "LogicalResourceId": "MySNSTopic"
    },
    "Destination": {
      "StackName": "Stack2",
      "LogicalResourceId": "MyLambdaSNSTopic"
    }
  }
]
```

2. Utilice el comando [describe-stack-refactor](#) para asegurarse de que `Status` sea `CREATE_COMPLETE`. Esto verifica que la validación esté completa.

```
aws cloudformation describe-stack-refactor \
  --stack-refactor-id 9c384f70-4e07-4ed7-a65d-fee5eb430841
```

Ejemplo de código de salida:

```
{
```

```

    "StackRefactorId": "9c384f70-4e07-4ed7-a65d-fee5eb430841",
    "StackIds": [
        "arn:aws:cloudformation:us-east-1:123456789012:stack/Stack1/3e6a1ff0-94b1-11f0-aa6f-0a88d2e03acf",
        "arn:aws:cloudformation:us-east-1:123456789012:stack/Stack2/5da91650-94b1-11f0-81cf-0a23500e151b"
    ],
    "ExecutionStatus": "AVAILABLE",
    "Status": "CREATE_COMPLETE"
}

```

3. Utilice el comando [list-stack-refactor-actions](#) para obtener una vista previa de las acciones específicas que se van a realizar.

```

aws cloudformation list-stack-refactor-actions \
  --stack-refactor-id 9c384f70-4e07-4ed7-a65d-fee5eb430841

```

Ejemplo de código de salida:

```

{
  "StackRefactorActions": [
    {
      "Action": "MOVE",
      "Entity": "RESOURCE",
      "PhysicalResourceId": "MyTestLambdaRole",
      "Description": "No configuration changes detected.",
      "Detection": "AUTO",
      "TagResources": [],
      "UntagResources": [],
      "ResourceMapping": {
        "Source": {
          "StackName": "arn:aws:cloudformation:us-east-1:123456789012:stack/Stack1/3e6a1ff0-94b1-11f0-aa6f-0a88d2e03acf",
          "LogicalResourceId": "MyLambdaRole"
        },
        "Destination": {
          "StackName": "arn:aws:cloudformation:us-east-1:123456789012:stack/Stack2/5da91650-94b1-11f0-81cf-0a23500e151b",
          "LogicalResourceId": "MyLambdaRole"
        }
      }
    },
    {

```

```

        "Action": "MOVE",
        "Entity": "RESOURCE",
        "PhysicalResourceId": "MyTestFunction",
        "Description": "Resource configuration changes will be validated during
refactor execution.",
        "Detection": "AUTO",
        "TagResources": [
            {
                "Key": "aws:cloudformation:stack-name",
                "Value": "Stack2"
            },
            {
                "Key": "aws:cloudformation:logical-id",
                "Value": "MyFunction"
            },
            {
                "Key": "aws:cloudformation:stack-id",
                "Value": "arn:aws:cloudformation:us-east-1:123456789012:stack/
Stack2/5da91650-94b1-11f0-81cf-0a23500e151b"
            }
        ],
        "UntagResources": [
            "aws:cloudformation:stack-name",
            "aws:cloudformation:logical-id",
            "aws:cloudformation:stack-id"
        ],
        "ResourceMapping": {
            "Source": {
                "StackName": "arn:aws:cloudformation:us-
east-1:123456789012:stack/Stack1/3e6a1ff0-94b1-11f0-aa6f-0a88d2e03acf",
                "LogicalResourceId": "MyFunction"
            },
            "Destination": {
                "StackName": "arn:aws:cloudformation:us-
east-1:123456789012:stack/Stack2/5da91650-94b1-11f0-81cf-0a23500e151b",
                "LogicalResourceId": "MyFunction"
            }
        }
    }
}
]
}

```

4. Tras revisar y confirmar los cambios, utilice el comando [execute-stack-refactor](#) para completar la operación de refactorización de pilas.


```
aws cloudformation execute-stack-refactor \  
--stack-refactor-id 9c384f70-4e07-4ed7-a65d-fee5eb430841
```

5. Utilice el comando [describe-stack-refactor](#) para supervisar el estado de la ejecución.

```
aws cloudformation describe-stack-refactor \  
--stack-refactor-id 9c384f70-4e07-4ed7-a65d-fee5eb430841
```

Ejemplo de código de salida:

```
{  
  "StackRefactorId": "9c384f70-4e07-4ed7-a65d-fee5eb430841",  
  "StackIds": [  
    "arn:aws:cloudformation:us-east-1:123456789012:stack/  
Stack1/3e6a1ff0-94b1-11f0-aa6f-0a88d2e03acf",  
    "arn:aws:cloudformation:us-east-1:123456789012:stack/  
Stack2/5da91650-94b1-11f0-81cf-0a23500e151b"  
  ],  
  "ExecutionStatus": "SUCCEEDED",  
  "Status": "COMPLETE"  
}
```

Limitaciones de recursos

- La refactorización de pilas solo admite los tipos de recursos con un provisioningType de FULLY_MUTABLE, lo que puede comprobar con el comando [describe-type](#).
- CloudFormation validará la elegibilidad de los recursos durante la creación de la refactorización e informará de cualquier recurso no compatible en la salida del comando [describe-stack-refactor](#).
- Los siguientes recursos no son compatibles con la refactorización de pilas:
 - AWS::ACMPCA::Certificate
 - AWS::ACMPCA::CertificateAuthority
 - AWS::ACMPCA::CertificateAuthorityActivation
 - AWS::ApiGateway::BasePathMapping
 - AWS::ApiGateway::Method
 - AWS::AppConfig::ConfigurationProfile

- `AWS::AppConfig::Deployment`
- `AWS::AppConfig::Environment`
- `AWS::AppConfig::Extension`
- `AWS::AppConfig::ExtensionAssociation`
- `AWS::AppStream::DirectoryConfig`
- `AWS::AppStream::StackFleetAssociation`
- `AWS::AppStream::StackUserAssociation`
- `AWS::AppStream::User`
- `AWS::BackupGateway::Hypervisor`
- `AWS::CertificateManager::Certificate`
- `AWS::CloudFormation::CustomResource`
- `AWS::CloudFormation::Macro`
- `AWS::CloudFormation::WaitCondition`
- `AWS::CloudFormation::WaitConditionHandle`
- `AWS::CodeDeploy::DeploymentGroup`
- `AWS::CodePipeline::CustomActionType`
- `AWS::Cognito::UserPoolRiskConfigurationAttachment`
- `AWS::Cognito::UserPoolUICustomizationAttachment`
- `AWS::Cognito::UserPoolUserToGroupAttachment`
- `AWS::Config::ConfigRule`
- `AWS::Config::ConfigurationRecorder`
- `AWS::Config::DeliveryChannel`
- `AWS::DataBrew::Dataset`
- `AWS::DataBrew::Job`
- `AWS::DataBrew::Project`
- `AWS::DataBrew::Recipe`
- `AWS::DataBrew::Ruleset`
- `AWS::DataBrew::Schedule`
- `AWS::DataZone::DataSource`
- `AWS::DataZone::Environment`

- `AWS::DataZone::EnvironmentBlueprintConfiguration`
- `AWS::DataZone::EnvironmentProfile`
- `AWS::DataZone::Project`
- `AWS::DataZone::SubscriptionTarget`
- `AWS::DirectoryService::MicrosoftAD`
- `AWS::DynamoDB::GlobalTable`
- `AWS::EC2::CustomerGateway`
- `AWS::EC2::EIP`
- `AWS::EC2::LaunchTemplate`
- `AWS::EC2::NetworkInterfacePermission`
- `AWS::EC2::PlacementGroup`
- `AWS::EC2::SpotFleet`
- `AWS::EC2::VPCDHCPOptionsAssociation`
- `AWS::EC2::VolumeAttachment`
- `AWS::EMR::Cluster`
- `AWS::EMR::InstanceFleetConfig`
- `AWS::EMR::InstanceGroupConfig`
- `AWS::ElastiCache::CacheCluster`
- `AWS::ElastiCache::ReplicationGroup`
- `AWS::ElastiCache::SecurityGroup`
- `AWS::ElastiCache::SecurityGroupIngress`
- `AWS::ElasticBeanstalk::ConfigurationTemplate`
- `AWS::ElasticLoadBalancing::LoadBalancer`
- `AWS::ElasticLoadBalancingV2::ListenerCertificate`
- `AWS::Elasticsearch::Domain`
- `AWS::FIS::ExperimentTemplate`
- `AWS::Glue::Schema`
- `AWS::GuardDuty::IPSet`
- `AWS::GuardDuty::PublishingDestination`
- `AWS::GuardDuty::ThreatIntelSet`

- `AWS::IAM::AccessKey`
- `AWS::IAM::UserToGroupAddition`
- `AWS::ImageBuilder::Component`
- `AWS::IoT::PolicyPrincipalAttachment`
- `AWS::IoT::ThingPrincipalAttachment`
- `AWS::IoTFleetWise::Campaign`
- `AWS::IoTWireless::WirelessDeviceImportTask`
- `AWS::Lambda::EventInvokeConfig`
- `AWS::Lex::BotVersion`
- `AWS::M2::Application`
- `AWS::MSK::Configuration`
- `AWS::MSK::ServerlessCluster`
- `AWS::Maester::DocumentType`
- `AWS::MediaTailor::Channel`
- `AWS::NeptuneGraph::PrivateGraphEndpoint`
- `AWS::Omics::AnnotationStore`
- `AWS::Omics::ReferenceStore`
- `AWS::Omics::SequenceStore`
- `AWS::OpenSearchServerless::Collection`
- `AWS::OpsWorks::App`
- `AWS::OpsWorks::ElasticLoadBalancerAttachment`
- `AWS::OpsWorks::Instance`
- `AWS::OpsWorks::Layer`
- `AWS::OpsWorks::Stack`
- `AWS::OpsWorks::UserProfile`
- `AWS::OpsWorks::Volume`
- `AWS::PCAConnectorAD::Connector`
- `AWS::PCAConnectorAD::DirectoryRegistration`
- `AWS::PCAConnectorAD::Template`
- `AWS::PCAConnectorAD::TemplateGroupAccessControlEntry`

- AWS::Panorama::PackageVersion
- AWS::QuickSight::Theme
- AWS::RDS::DBSecurityGroup
- AWS::RDS::DBSecurityGroupIngress
- AWS::Redshift::ClusterSecurityGroup
- AWS::Redshift::ClusterSecurityGroupIngress
- AWS::RefactorSpaces::Environment
- AWS::RefactorSpaces::Route
- AWS::RefactorSpaces::Service
- AWS::RoboMaker::RobotApplication
- AWS::RoboMaker::SimulationApplication
- AWS::Route53::RecordSet
- AWS::Route53::RecordSetGroup
- AWS::SDB::Domain
- AWS::SageMaker::InferenceComponen
- AWS::ServiceCatalog::PortfolioPrincipalAssociation
- AWS::ServiceCatalog::PortfolioProductAssociation
- AWS::ServiceCatalog::PortfolioShare
- AWS::ServiceCatalog::TagOptionAssociation
- AWS::ServiceCatalogAppRegistry::AttributeGroupAssociation
- AWS::ServiceCatalogAppRegistry::ResourceAssociation
- AWS::StepFunctions::StateMachineVersion
- AWS::Synthetics::Canary
- AWS::VoiceID::Domain
- AWS::WAF::ByteMatchSet
- AWS::WAF::IPSet
- AWS::WAF::Rule
- AWS::WAF::SizeConstraintSet
- AWS::WAF::SqlInjectionMatchSet
- AWS::WAF::WebACL

- `AWS::WAF::XssMatchSet`
- `AWS::WAFv2::IPSet`
- `AWS::WAFv2::RegexPatternSet`
- `AWS::WAFv2::RuleGroup`
- `AWS::WAFv2::WebACL`
- `AWS::WorkSpaces::Workspace`

Tipo de recurso compatible

En la siguiente tabla se enumeran los tipos de recursos de AWS que actualmente admiten las operaciones de importación, detección de desviaciones y generadores de infraestructura como código (IaC). El nombre de cada tipo de recurso contiene un vínculo a su tema de referencia correspondiente en la [Guía de referencia de plantillas de CloudFormation](#).

Un recurso que admite la importación de recursos también podría admitir la autoimportación. Para obtener más información, consulte [Cómo importar recursos de AWS a una pila de CloudFormation](#).

No es una lista exhaustiva de recursos de AWS. Si un tipo de recurso específico no aparece en la lista, es probable que no se pueda acceder a este a través de la API de control de nube de AWS. Para obtener más información, consulte [Tipos de recursos compatibles con la API de control en la nube](#) en la Guía del usuario de la API de control en la nube. Cada servicio de AWS determina de forma independiente a qué tipos de recursos se puede acceder a través de la API de Cloud Control.

Además, CloudFormation admite operaciones de importaciones y detección de desviaciones para tipos de recursos privados que son aprovisionables; es decir, cuyo tipo de aprovisionamiento es `FULLY_MUTABLE` o `IMMUTABLE`. Para importar o detectar desviaciones en un tipo de recurso privado, primero debe registrar la versión predeterminada de dicho tipo en la cuenta y asegurarse de que sea aprovisionable. Para obtener más información, consulte [Uso de extensiones privadas de terceros que se han compartido con usted](#).

Tenga en cuenta que el generador de IaC solo admite recursos de AWS compatibles con la API de Cloud Control en la región.

Para acceder mediante programación a la información sobre los tipos de recursos aprovisionables, tanto públicos como privados, puede utilizar la API de control de nube de AWS. Para obtener más información, consulte [Determinar si un tipo de recurso admite la API de Cloud Control](#) en la Guía del usuario de la API de Cloud Control.

Para comenzar a utilizar la importación, la detección de desviaciones o el generador de laC, estos son algunos temas útiles que conviene repasar:

- [Cómo importar recursos de AWS a una pila de CloudFormation](#)
- [Detectar cambios de configuración no administrados en pilas y recursos con detección de derivación](#)
- [Generar plantillas de los recursos existentes con el generador laC](#)

Recurso	Importación	Detección de desviaciones	Generador de laC
AWS::ACMPCA::Certificate	 Sí	 No	 No
AWS::ACMPCA::CertificateAuthority	 Sí	 Sí	 Sí
AWS::ACMPCA::CertificateAuthorityActivation	 Sí	 No	 No
AWS::ACMPCA::Permission	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::AIOps::InvestigationGroup</u>	 Sí	 Sí	 No
<u>AWS::APS::AnomalyDetector</u>	 Sí	 Sí	 No
<u>AWS::APS::ResourcePolicy</u>	 Sí	 Sí	 No
<u>AWS::APS::RuleGroupsNamespace</u>	 Sí	 Sí	 Sí
<u>AWS::APS::Scraper</u>	 Sí	 Sí	 No
<u>AWS::APS::Workspace</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::ARCRegionSwitch::Plan</u>	 Sí	 Sí	 No
<u>AWS::ARCZonalShift::AutoshiftObserverNotificationStatus</u>	 Sí	 Sí	 No
<u>AWS::ARCZonalShift::ZonalAutoshiftConfiguration</u>	 Sí	 Sí	 No
<u>AWS::AccessAnalyzer::Analyzer</u>	 Sí	 Sí	 Sí
<u>AWS::AmazonMQ::Broker</u>	 Sí	 Sí	 No
<u>AWS::AmazonMQ::Configuration</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::Amplify::App</u>	 Sí	 Sí	 Sí
<u>AWS::Amplify::Branch</u>	 Sí	 Sí	 Sí
<u>AWS::Amplify::Domain</u>	 Sí	 Sí	 Sí
<u>AWS::AmplifyUIBuilder::Component</u>	 Sí	 Sí	 Sí
<u>AWS::AmplifyUIBuilder::Form</u>	 Sí	 Sí	 Sí
<u>AWS::AmplifyUIBuilder::Theme</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::ApiGateway::Account</u>	 Sí	 Sí	 No
<u>AWS::ApiGateway::ApiKey</u>	 Sí	 Sí	 Sí
<u>AWS::ApiGateway::Authorizer</u>	 Sí	 Sí	 No
<u>AWS::ApiGateway::BasePathMapping</u>	 Sí	 Sí	 No
<u>AWS::ApiGateway::BasePathMappingV2</u>	 Sí	 Sí	 No
<u>AWS::ApiGateway::ClientCertificate</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::ApiGateway::Deployment</u>	 Sí	 Sí	 Sí
<u>AWS::ApiGateway::DocumentationPart</u>	 Sí	 Sí	 No
<u>AWS::ApiGateway::DocumentationVersion</u>	 Sí	 Sí	 Sí
<u>AWS::ApiGateway::DomainName</u>	 Sí	 Sí	 No
<u>AWS::ApiGateway::DomainNameAccessAssociation</u>	 Sí	 Sí	 No
<u>AWS::ApiGateway::DomainNameV2</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::ApiGateway::GatewayResponse</u>	 Sí	 Sí	 No
<u>AWS::ApiGateway::Method</u>	 Sí	 Sí	 No
<u>AWS::ApiGateway::Model</u>	 Sí	 Sí	 No
<u>AWS::ApiGateway::RequestValidator</u>	 Sí	 Sí	 No
<u>AWS::ApiGateway::Resource</u>	 Sí	 Sí	 No
<u>AWS::ApiGateway::RestApi</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::ApiGateway::Stage</u>	 Sí	 No	 Sí
<u>AWS::ApiGateway::UsagePlan</u>	 Sí	 Sí	 Sí
<u>AWS::ApiGateway::UsagePlanKey</u>	 Sí	 No	 Sí
<u>AWS::ApiGateway::VpcLink</u>	 Sí	 Sí	 No
<u>AWS::ApiGatewayV2::Api</u>	 Sí	 Sí	 Sí
<u>AWS::ApiGatewayV2::ApiMapping</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::ApiGatewayV2::Authorizer</u>	 Sí	 Sí	 No
<u>AWS::ApiGatewayV2::Deployment</u>	 Sí	 Sí	 Sí
<u>AWS::ApiGatewayV2::DomainName</u>	 Sí	 Sí	 Sí
<u>AWS::ApiGatewayV2::Integration</u>	 Sí	 Sí	 No
<u>AWS::ApiGatewayV2::IntegrationResponse</u>	 Sí	 Sí	 No
<u>AWS::ApiGatewayV2::Model</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::ApiGatewayV2::Route</u>	 Sí	 Sí	 No
<u>AWS::ApiGatewayV2::RouteResponse</u>	 Sí	 Sí	 No
<u>AWS::ApiGatewayV2::RoutingRule</u>	 Sí	 Sí	 No
<u>AWS::ApiGatewayV2::VpcLink</u>	 Sí	 Sí	 Sí
<u>AWS::AppConfig::Application</u>	 Sí	 Sí	 Sí
<u>AWS::AppConfig::ConfigurationProfile</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::AppConfig::Deployment</u>	 Sí	 Sí	 No
<u>AWS::AppConfig::DeploymentStrategy</u>	 Sí	 Sí	 No
<u>AWS::AppConfig::Environment</u>	 Sí	 Sí	 No
<u>AWS::AppConfig::Extension</u>	 Sí	 Sí	 No
<u>AWS::AppConfig::ExtensionAssociation</u>	 Sí	 Sí	 No
<u>AWS::AppConfig::HostedConfigurationVersion</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::AppFlow::Connector</u>	 Sí	 Sí	 Sí
<u>AWS::AppFlow::ConnectorProfile</u>	 Sí	 Sí	 Sí
<u>AWS::AppFlow::Flow</u>	 Sí	 Sí	 Sí
<u>AWS::AppIntegrations::Application</u>	 Sí	 Sí	 No
<u>AWS::AppIntegrations::DataIntegration</u>	 Sí	 Sí	 Sí
<u>AWS::AppIntegrations::EventIntegration</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::AppRunner::AutoScalingConfiguration</u>	 Sí	 Sí	 No
<u>AWS::AppRunner::ObservabilityConfiguration</u>	 Sí	 Sí	 No
<u>AWS::AppRunner::Service</u>	 Sí	 Sí	 No
<u>AWS::AppRunner::VpcConnector</u>	 Sí	 Sí	 No
<u>AWS::AppRunner::VpcIngressConnection</u>	 Sí	 Sí	 No
<u>AWS::AppStream::AppBlock</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::AppStream::AppBlockBuilder</u>	 Sí	 Sí	 No
<u>AWS::AppStream::Application</u>	 Sí	 Sí	 No
<u>AWS::AppStream::ApplicationEntitlementAssociation</u>	 Sí	 Sí	 No
<u>AWS::AppStream::ApplicationFleetAssociation</u>	 Sí	 Sí	 No
<u>AWS::AppStream::DirectoryConfig</u>	 Sí	 Sí	 Sí
<u>AWS::AppStream::Entitlement</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::AppStream::ImageBuilder</u>	 Sí	 Sí	 Sí
<u>AWS::AppSync::Api</u>	 Sí	 Sí	 No
<u>AWS::AppSync::ChannelNamespace</u>	 Sí	 Sí	 No
<u>AWS::AppSync::DataSource</u>	 Sí	 Sí	 No
<u>AWS::AppSync::DomainName</u>	 Sí	 Sí	 Sí
<u>AWS::AppSync::DomainNameApiAssociation</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::AppSync::FunctionConfiguration</u>	 Sí	 Sí	 No
<u>AWS::AppSync::GraphQLApi</u>	 No	 Sí	 No
<u>AWS::AppSync::Resolver</u>	 Sí	 Sí	 No
<u>AWS::AppSync::SourceApiAssociation</u>	 Sí	 Sí	 Sí
<u>AWS::AppTest::TestCase</u>	 Sí	 Sí	 No
<u>AWS::ApplicationAutoScaling::ScalableTarget</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::ApplicationAutoScaling::ScalingPolicy</u>	 Sí	 No	 No
<u>AWS::ApplicationInsights::Application</u>	 Sí	 Sí	 No
<u>AWS::ApplicationSignals::Discovery</u>	 Sí	 Sí	 No
<u>AWS::ApplicationSignals::GroupingConfiguration</u>	 Sí	 Sí	 No
<u>AWS::ApplicationSignals::ServiceLevelObjective</u>	 Sí	 Sí	 No
<u>AWS::Athena::CapacityReservation</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::Athena::DataCatalog</u>	 Sí	 Sí	 Sí
<u>AWS::Athena::NamedQuery</u>	 Sí	 Sí	 Sí
<u>AWS::Athena::PreparedStatement</u>	 Sí	 Sí	 Sí
<u>AWS::Athena::WorkGroup</u>	 Sí	 Sí	 Sí
<u>AWS::AuditManager::Assessment</u>	 Sí	 No	 No
<u>AWS::AutoScaling::AutoScalingGroup</u>	 Sí	 No	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::AutoScaling::LaunchConfiguration</u>	 Sí	 Sí	 Sí
<u>AWS::AutoScaling::LifecycleHook</u>	 Sí	 Sí	 Sí
<u>AWS::AutoScaling::ScalingPolicy</u>	 Sí	 Sí	 Sí
<u>AWS::AutoScaling::ScheduledAction</u>	 Sí	 Sí	 Sí
<u>AWS::AutoScaling::WarmPool</u>	 Sí	 Sí	 No
<u>AWS::B2BI::Capability</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::B2BI::Partnership</u>	 Sí	 Sí	 No
<u>AWS::B2BI::Profile</u>	 Sí	 Sí	 No
<u>AWS::B2BI::Transformer</u>	 Sí	 Sí	 No
<u>AWS::BCMDDataExports::Export</u>	 Sí	 Sí	 No
<u>AWS::Backup::BackupPlan</u>	 Sí	 Sí	 No
<u>AWS::Backup::BackupSelection</u>	 Sí	 No	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::Backup::BackupVault</u>	 Sí	 Sí	 No
<u>AWS::Backup::Framework</u>	 Sí	 Sí	 Sí
<u>AWS::Backup::LogicallyAirGappedBackupVault</u>	 Sí	 Sí	 No
<u>AWS::Backup::ReportPlan</u>	 Sí	 Sí	 Sí
<u>AWS::Backup::RestoreTestingPlan</u>	 Sí	 Sí	 No
<u>AWS::Backup::RestoreTestingSelection</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::BackupGateway::Hypervisor</u>	 Sí	 Sí	 No
<u>AWS::Batch::ComputeEnvironment</u>	 Sí	 Sí	 Sí
<u>AWS::Batch::ConsumableResource</u>	 Sí	 Sí	 No
<u>AWS::Batch::JobDefinition</u>	 Sí	 Sí	 No
<u>AWS::Batch::JobQueue</u>	 Sí	 Sí	 Sí
<u>AWS::Batch::SchedulingPolicy</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::Batch::ServiceEnvironment</u>	 Sí	 Sí	 No
<u>AWS::Bedrock::Agent</u>	 Sí	 Sí	 Sí
<u>AWS::Bedrock::AgentAlias</u>	 Sí	 Sí	 Sí
<u>AWS::Bedrock::ApplicationInferenceProfile</u>	 Sí	 Sí	 Sí
<u>AWS::Bedrock::AutomatedReasoningPolicy</u>	 Sí	 Sí	 No
<u>AWS::Bedrock::AutomatedReasoningPolicyVersion</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::Bedrock::Blueprint</u>	 Sí	 Sí	 Sí
<u>AWS::Bedrock::DataAutomationProject</u>	 Sí	 Sí	 Sí
<u>AWS::Bedrock::DataSource</u>	 Sí	 Sí	 Sí
<u>AWS::Bedrock::Flow</u>	 Sí	 Sí	 Sí
<u>AWS::Bedrock::FlowAlias</u>	 Sí	 Sí	 Sí
<u>AWS::Bedrock::FlowVersion</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::Bedrock::Guardrail</u>	 Sí	 Sí	 Sí
<u>AWS::Bedrock::GuardrailVersion</u>	 Sí	 Sí	 No
<u>AWS::Bedrock::IntelligentPromptRouter</u>	 Sí	 Sí	 Sí
<u>AWS::Bedrock::KnowledgeBase</u>	 Sí	 No	 Sí
<u>AWS::Bedrock::Prompt</u>	 Sí	 Sí	 Sí
<u>AWS::Bedrock::PromptVersion</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::BedrockAgentCore::BrowserCustom</u>	 Sí	 Sí	 No
<u>AWS::BedrockAgentCore::CodeInterpreterCustom</u>	 Sí	 Sí	 No
<u>AWS::BedrockAgentCore::Gateway</u>	 Sí	 Sí	 No
<u>AWS::BedrockAgentCore::GatewayTarget</u>	 Sí	 Sí	 No
<u>AWS::BedrockAgentCore::Memory</u>	 Sí	 Sí	 No
<u>AWS::BedrockAgentCore::Runtime</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
AWS::BedrockAgentCore::RuntimeEndpoint	 Sí	 Sí	 No
AWS::BedrockAgentCore::WorkloadIdentity	 Sí	 Sí	 No
AWS::Billing::BillingView	 Sí	 Sí	 No
AWS::BillingConductor::BillingGroup	 Sí	 Sí	 Sí
AWS::BillingConductor::CustomLineItem	 Sí	 Sí	 Sí
AWS::BillingConductor::PricingPlan	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::BillingConductor::PricingRule</u>	 Sí	 Sí	 Sí
<u>AWS::Budgets::BudgetsAction</u>	 Sí	 Sí	 Sí
<u>AWS::CE::AnomalyMonitor</u>	 Sí	 Sí	 No
<u>AWS::CE::AnomalySubscription</u>	 Sí	 Sí	 No
<u>AWS::CE::CostCategory</u>	 Sí	 Sí	 Sí
<u>AWS::CUR::ReportDefinition</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::Cases::CaseRule</u>	 Sí	 Sí	 No
<u>AWS::Cases::Domain</u>	 Sí	 Sí	 No
<u>AWS::Cases::Field</u>	 Sí	 Sí	 No
<u>AWS::Cases::Layout</u>	 Sí	 Sí	 No
<u>AWS::Cases::Template</u>	 Sí	 Sí	 No
<u>AWS::Cassandra::Keyspace</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::Cassandra::Table</u>	 Sí	 Sí	 Sí
<u>AWS::Cassandra::Type</u>	 Sí	 Sí	 No
<u>AWS::CertificateManager::Account</u>	 Sí	 Sí	 No
<u>AWS::Chatbot::CustomAction</u>	 Sí	 Sí	 No
<u>AWS::Chatbot::MicrosoftTeamsChannelConfiguration</u>	 Sí	 Sí	 Sí
<u>AWS::Chatbot::SlackChannelConfiguration</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::CleanRooms::AnalysisTemplate</u>	 Sí	 Sí	 Sí
<u>AWS::CleanRooms::Collaboration</u>	 Sí	 Sí	 Sí
<u>AWS::CleanRooms::ConfiguredTable</u>	 Sí	 Sí	 Sí
<u>AWS::CleanRooms::ConfiguredTableAssociation</u>	 Sí	 Sí	 Sí
<u>AWS::CleanRooms::IdMappingTable</u>	 Sí	 Sí	 No
<u>AWS::CleanRooms::IdNamespaceAssociation</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::CleanRooms::Membership</u>	 Sí	 Sí	 Sí
<u>AWS::CleanRooms::PrivacyBudgetTemplate</u>	 Sí	 Sí	 No
<u>AWS::CleanRoomsML::TrainingDataset</u>	 Sí	 Sí	 No
<u>AWS::CloudFormation::GuardHook</u>	 Sí	 Sí	 No
<u>AWS::CloudFormation::HookDefaultVersion</u>	 Sí	 Sí	 Sí
<u>AWS::CloudFormation::HookTypeConfig</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::CloudFormation::HookVersion</u>	 Sí	 Sí	 Sí
<u>AWS::CloudFormation::LambdaHook</u>	 Sí	 Sí	 No
<u>AWS::CloudFormation::ModuleDefaultVersion</u>	 Sí	 Sí	 No
<u>AWS::CloudFormation::ModuleVersion</u>	 Sí	 Sí	 No
<u>AWS::CloudFormation::PublicTypeVersion</u>	 Sí	 Sí	 No
<u>AWS::CloudFormation::Publisher</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::CloudFormation::ResourceDefaultVersion</u>	 Sí	 Sí	 No
<u>AWS::CloudFormation::ResourceVersion</u>	 Sí	 Sí	 No
<u>AWS::CloudFormation::Stack</u>	 Sí	 Sí	 No
<u>AWS::CloudFormation::StackSet</u>	 Sí	 Sí	 No
<u>AWS::CloudFormation::TypeActivation</u>	 Sí	 Sí	 No
<u>AWS::CloudFront::AnycastIpList</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::CloudFront::CachePolicy</u>	 Sí	 Sí	 Sí
<u>AWS::CloudFront::CloudFrontOriginAccessIdentity</u>	 Sí	 Sí	 Sí
<u>AWS::CloudFront::ConnectionFunction</u>	 Sí	 Sí	 No
<u>AWS::CloudFront::ConnectionGroup</u>	 Sí	 Sí	 No
<u>AWS::CloudFront::ContinuousDeploymentPolicy</u>	 Sí	 Sí	 Sí
<u>AWS::CloudFront::Distribution</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::CloudFront::DistributionTenant</u>	 Sí	 Sí	 No
<u>AWS::CloudFront::Function</u>	 Sí	 Sí	 No
<u>AWS::CloudFront::KeyGroup</u>	 Sí	 Sí	 Sí
<u>AWS::CloudFront::KeyValueStore</u>	 Sí	 Sí	 No
<u>AWS::CloudFront::MonitoringSubscription</u>	 Sí	 Sí	 No
<u>AWS::CloudFront::OriginAccessControl</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::CloudFront::OriginRequestPolicy</u>	 Sí	 Sí	 Sí
<u>AWS::CloudFront::PublicKey</u>	 Sí	 Sí	 Sí
<u>AWS::CloudFront::RealtimeLogConfig</u>	 Sí	 Sí	 Sí
<u>AWS::CloudFront::ResponseHeadersPolicy</u>	 Sí	 Sí	 No
<u>AWS::CloudFront::TrustStore</u>	 Sí	 Sí	 No
<u>AWS::CloudFront::VpcOrigin</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::CloudTrail::Channel</u>	 Sí	 Sí	 No
<u>AWS::CloudTrail::Dashboard</u>	 Sí	 Sí	 No
<u>AWS::CloudTrail::EventDataStore</u>	 Sí	 Sí	 Sí
<u>AWS::CloudTrail::ResourcePolicy</u>	 Sí	 Sí	 No
<u>AWS::CloudTrail::Trail</u>	 Sí	 Sí	 Sí
<u>AWS::CloudWatch::Alarm</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::CloudWatch::CompositeAlarm</u>	 Sí	 Sí	 Sí
<u>AWS::CloudWatch::Dashboard</u>	 Sí	 Sí	 No
<u>AWS::CloudWatch::MetricStream</u>	 Sí	 Sí	 No
<u>AWS::CodeArtifact::Domain</u>	 Sí	 Sí	 Sí
<u>AWS::CodeArtifact::PackageGroup</u>	 Sí	 Sí	 No
<u>AWS::CodeArtifact::Repository</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::CodeBuild::Fleet</u>	 Sí	 Sí	 No
<u>AWS::CodeConnections::Connection</u>	 Sí	 Sí	 No
<u>AWS::CodeDeploy::Application</u>	 Sí	 Sí	 Sí
<u>AWS::CodeDeploy::DeploymentConfig</u>	 Sí	 Sí	 No
<u>AWS::CodeGuruProfiler::ProfilingGroup</u>	 Sí	 Sí	 Sí
<u>AWS::CodeGuruReviewer::RepositoryAssociation</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::CodePipeline::CustomActionType</u>	 Sí	 Sí	 No
<u>AWS::CodePipeline::Pipeline</u>	 Sí	 Sí	 No
<u>AWS::CodePipeline::Webhook</u>	 Sí	 Sí	 No
<u>AWS::CodeStarConnections::Connection</u>	 Sí	 Sí	 Sí
<u>AWS::CodeStarConnections::Repository Link</u>	 Sí	 Sí	 No
<u>AWS::CodeStarConnections::SyncConfiguration</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::CodeStarNotifications::NotificationRule</u>	 Sí	 Sí	 Sí
<u>AWS::Cognito::IdentityPool</u>	 Sí	 Sí	 No
<u>AWS::Cognito::IdentityPoolPrincipalTag</u>	 Sí	 Sí	 Sí
<u>AWS::Cognito::IdentityPoolRoleAttachment</u>	 Sí	 Sí	 No
<u>AWS::Cognito::LogDeliveryConfiguration</u>	 Sí	 No	 No
<u>AWS::Cognito::ManagedLoginBranding</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::Cognito::Terms</u>	 Sí	 Sí	 No
<u>AWS::Cognito::UserPool</u>	 Sí	 No	 No
<u>AWS::Cognito::UserPoolClient</u>	 Sí	 Sí	 No
<u>AWS::Cognito::UserPoolDomain</u>	 Sí	 Sí	 No
<u>AWS::Cognito::UserPoolGroup</u>	 Sí	 Sí	 Sí
<u>AWS::Cognito::UserPoolIdentityProvider</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::Cognito::UserPoolResourceServer</u>	 Sí	 Sí	 No
<u>AWS::Cognito::UserPoolRiskConfigurationAttachment</u>	 Sí	 Sí	 No
<u>AWS::Cognito::UserPoolUICustomizationAttachment</u>	 Sí	 Sí	 No
<u>AWS::Cognito::UserPoolUser</u>	 Sí	 No	 No
<u>AWS::Cognito::UserPoolUserToGroupAttachment</u>	 Sí	 Sí	 No
<u>AWS::Comprehend::DocumentClassifier</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::Comprehend::Flywheel</u>	 Sí	 Sí	 Sí
<u>AWS::Config::AggregationAuthorization</u>	 Sí	 Sí	 Sí
<u>AWS::Config::ConfigRule</u>	 Sí	 Sí	 Sí
<u>AWS::Config::ConfigurationAggregator</u>	 Sí	 Sí	 Sí
<u>AWS::Config::ConformancePack</u>	 Sí	 Sí	 Sí
<u>AWS::Config::OrganizationConformancePack</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::Config::StoredQuery</u>	 Sí	 Sí	 Sí
<u>AWS::Connect::AgentStatus</u>	 Sí	 Sí	 Sí
<u>AWS::Connect::ApprovedOrigin</u>	 Sí	 Sí	 Sí
<u>AWS::Connect::ContactFlow</u>	 Sí	 Sí	 Sí
<u>AWS::Connect::ContactFlowModule</u>	 Sí	 Sí	 Sí
<u>AWS::Connect::ContactFlowVersion</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::Connect::DataTable</u>	 Sí	 Sí	 No
<u>AWS::Connect::DataTableAttribute</u>	 Sí	 Sí	 No
<u>AWS::Connect::DataTableRecord</u>	 Sí	 Sí	 No
<u>AWS::Connect::EmailAddress</u>	 Sí	 Sí	 Sí
<u>AWS::Connect::EvaluationForm</u>	 Sí	 Sí	 Sí
<u>AWS::Connect::HoursOfOperation</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::Connect::Instance</u>	 Sí	 Sí	 Sí
<u>AWS::Connect::InstanceStorageConfig</u>	 Sí	 Sí	 Sí
<u>AWS::Connect::IntegrationAssociation</u>	 Sí	 Sí	 Sí
<u>AWS::Connect::PhoneNumber</u>	 Sí	 Sí	 Sí
<u>AWS::Connect::PredefinedAttribute</u>	 Sí	 Sí	 Sí
<u>AWS::Connect::Prompt</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::Connect::Queue</u>	 Sí	 Sí	 Sí
<u>AWS::Connect::QuickConnect</u>	 Sí	 Sí	 Sí
<u>AWS::Connect::RoutingProfile</u>	 Sí	 Sí	 Sí
<u>AWS::Connect::Rule</u>	 Sí	 Sí	 No
<u>AWS::Connect::SecurityKey</u>	 Sí	 Sí	 Sí
<u>AWS::Connect::SecurityProfile</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::Connect::TaskTemplate</u>	 Sí	 Sí	 Sí
<u>AWS::Connect::TrafficDistributionGroup</u>	 Sí	 Sí	 Sí
<u>AWS::Connect::User</u>	 Sí	 Sí	 Sí
<u>AWS::Connect::UserHierarchyGroup</u>	 Sí	 Sí	 Sí
<u>AWS::Connect::UserHierarchyStructure</u>	 Sí	 Sí	 No
<u>AWS::Connect::View</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::Connect::ViewVersion</u>	 Sí	 Sí	 Sí
<u>AWS::Connect::Workspace</u>	 Sí	 Sí	 No
<u>AWS::ConnectCampaigns::Campaign</u>	 Sí	 Sí	 Sí
<u>AWS::ConnectCampaignsV2::Campaign</u>	 Sí	 Sí	 Sí
<u>AWS::ControlTower::EnabledBaseline</u>	 Sí	 Sí	 No
<u>AWS::ControlTower::EnabledControl</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::ControlTower::LandingZone</u>	 Sí	 Sí	 No
<u>AWS::CustomerProfiles::CalculatedAttributeDefinition</u>	 Sí	 Sí	 Sí
<u>AWS::CustomerProfiles::Domain</u>	 Sí	 Sí	 Sí
<u>AWS::CustomerProfiles::EventStream</u>	 Sí	 Sí	 Sí
<u>AWS::CustomerProfiles::EventTrigger</u>	 Sí	 Sí	 No
<u>AWS::CustomerProfiles::Integration</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::CustomerProfiles::ObjectType</u>	 Sí	 Sí	 Sí
<u>AWS::CustomerProfiles::SegmentDefinition</u>	 Sí	 Sí	 No
<u>AWS::DMS::DataMigration</u>	 Sí	 Sí	 No
<u>AWS::DMS::DataProvider</u>	 Sí	 Sí	 No
<u>AWS::DMS::InstanceProfile</u>	 Sí	 Sí	 No
<u>AWS::DMS::MigrationProject</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::DMS::ReplicationConfig</u>	 Sí	 Sí	 Sí
<u>AWS::DSQL::Cluster</u>	 Sí	 Sí	 No
<u>AWS::DataBrew::Dataset</u>	 Sí	 Sí	 Sí
<u>AWS::DataBrew::Job</u>	 Sí	 Sí	 Sí
<u>AWS::DataBrew::Project</u>	 Sí	 Sí	 Sí
<u>AWS::DataBrew::Recipe</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::DataBrew::Ruleset</u>	 Sí	 Sí	 Sí
<u>AWS::DataBrew::Schedule</u>	 Sí	 Sí	 Sí
<u>AWS::DataPipeline::Pipeline</u>	 Sí	 Sí	 Sí
<u>AWS::DataSync::Agent</u>	 Sí	 Sí	 Sí
<u>AWS::DataSync::LocationAzureBlob</u>	 Sí	 Sí	 Sí
<u>AWS::DataSync::LocationEFS</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::DataSync::LocationFSxLustre</u>	 Sí	 Sí	 Sí
<u>AWS::DataSync::LocationFSxONTAP</u>	 Sí	 Sí	 Sí
<u>AWS::DataSync::LocationFSxOpenZFS</u>	 Sí	 Sí	 Sí
<u>AWS::DataSync::LocationFSxWindows</u>	 Sí	 Sí	 Sí
<u>AWS::DataSync::LocationHDFS</u>	 Sí	 Sí	 Sí
<u>AWS::DataSync::LocationNFS</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::DataSync::LocationObjectStorage</u>	 Sí	 Sí	 Sí
<u>AWS::DataSync::LocationS3</u>	 Sí	 Sí	 Sí
<u>AWS::DataSync::LocationSMB</u>	 Sí	 Sí	 Sí
<u>AWS::DataSync::Task</u>	 Sí	 Sí	 Sí
<u>AWS::DataZone::Connection</u>	 Sí	 Sí	 No
<u>AWS::DataZone::DataSource</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::DataZone::Domain</u>	 Sí	 Sí	 No
<u>AWS::DataZone::DomainUnit</u>	 Sí	 Sí	 No
<u>AWS::DataZone::Environment</u>	 Sí	 Sí	 No
<u>AWS::DataZone::EnvironmentActions</u>	 Sí	 Sí	 No
<u>AWS::DataZone::EnvironmentBlueprintConfiguration</u>	 Sí	 Sí	 No
<u>AWS::DataZone::EnvironmentProfile</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::DataZone::FormType</u>	 Sí	 Sí	 No
<u>AWS::DataZone::GroupProfile</u>	 Sí	 Sí	 No
<u>AWS::DataZone::Owner</u>	 Sí	 Sí	 No
<u>AWS::DataZone::PolicyGrant</u>	 Sí	 Sí	 No
<u>AWS::DataZone::Project</u>	 Sí	 Sí	 No
<u>AWS::DataZone::ProjectMembership</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::DataZone::ProjectProfile</u>	 Sí	 Sí	 No
<u>AWS::DataZone::SubscriptionTarget</u>	 Sí	 Sí	 No
<u>AWS::DataZone::UserProfile</u>	 Sí	 Sí	 No
<u>AWS::Deadline::Farm</u>	 Sí	 Sí	 No
<u>AWS::Deadline::Fleet</u>	 Sí	 Sí	 No
<u>AWS::Deadline::LicenseEndpoint</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::Deadline::Limit</u>	 Sí	 Sí	 No
<u>AWS::Deadline::MeteredProduct</u>	 Sí	 Sí	 No
<u>AWS::Deadline::Monitor</u>	 Sí	 Sí	 No
<u>AWS::Deadline::Queue</u>	 Sí	 Sí	 No
<u>AWS::Deadline::QueueEnvironment</u>	 Sí	 Sí	 No
<u>AWS::Deadline::QueueFleetAssociation</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::Deadline::QueueLimitAssociation</u>	 Sí	 Sí	 No
<u>AWS::Deadline::StorageProfile</u>	 Sí	 Sí	 No
<u>AWS::Detective::Graph</u>	 Sí	 Sí	 Sí
<u>AWS::Detective::MemberInvitation</u>	 Sí	 Sí	 Sí
<u>AWS::Detective::OrganizationAdmin</u>	 Sí	 Sí	 Sí
<u>AWS::DevOpsAgent::AgentSpace</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::DevOpsAgent::Association</u>	 Sí	 Sí	 No
<u>AWS::DevOpsGuru::LogAnomalyDetectionIntegration</u>	 Sí	 Sí	 Sí
<u>AWS::DevOpsGuru::NotificationChannel</u>	 Sí	 Sí	 Sí
<u>AWS::DevOpsGuru::ResourceCollection</u>	 Sí	 Sí	 Sí
<u>AWS::DeviceFarm::DevicePool</u>	 Sí	 Sí	 No
<u>AWS::DeviceFarm::InstanceProfile</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::DeviceFarm::NetworkProfile</u>	 Sí	 Sí	 No
<u>AWS::DeviceFarm::Project</u>	 Sí	 Sí	 No
<u>AWS::DeviceFarm::TestGridProject</u>	 Sí	 Sí	 No
<u>AWS::DeviceFarm::VPCEConfiguration</u>	 Sí	 Sí	 No
<u>AWS::DirectoryService::SimpleAD</u>	 Sí	 Sí	 Sí
<u>AWS::DocDBElastic::Cluster</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::DynamoDB::GlobalTable</u>	 Sí	 Sí	 No
<u>AWS::DynamoDB::Table</u>	 Sí	 Sí	 Sí
<u>AWS::EC2::CapacityManagerDataExport</u>	 Sí	 Sí	 No
<u>AWS::EC2::CapacityReservation</u>	 Sí	 Sí	 Sí
<u>AWS::EC2::CapacityReservationFleet</u>	 Sí	 Sí	 Sí
<u>AWS::EC2::CarrierGateway</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::EC2::CustomerGateway</u>	 Sí	 Sí	 Sí
<u>AWS::EC2::DHCPOptions</u>	 Sí	 Sí	 Sí
<u>AWS::EC2::EC2Fleet</u>	 Sí	 Sí	 No
<u>AWS::EC2::EIP</u>	 Sí	 Sí	 Sí
<u>AWS::EC2::EIPAssociation</u>	 Sí	 No	 Sí
<u>AWS::EC2::EgressOnlyInternetGateway</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::EC2::EnclaveCertificateIamRoleAssociation</u>	 Sí	 Sí	 Sí
<u>AWS::EC2::FlowLog</u>	 Sí	 Sí	 Sí
<u>AWS::EC2::GatewayRouteTableAssociation</u>	 Sí	 Sí	 No
<u>AWS::EC2::Host</u>	 Sí	 Sí	 Sí
<u>AWS::EC2::IPAM</u>	 Sí	 Sí	 Sí
<u>AWS::EC2::IPAMAllocation</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::EC2::IPAMPool</u>	 Sí	 Sí	 Sí
<u>AWS::EC2::IPAMPoolCidr</u>	 Sí	 Sí	 Sí
<u>AWS::EC2::IPAMResourceDiscovery</u>	 Sí	 Sí	 Sí
<u>AWS::EC2::IPAMResourceDiscoveryAssociation</u>	 Sí	 Sí	 Sí
<u>AWS::EC2::IPAMScope</u>	 Sí	 Sí	 Sí
<u>AWS::EC2::Instance</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::EC2::InstanceConnectEndpoint</u>	 Sí	 Sí	 Sí
<u>AWS::EC2::InternetGateway</u>	 Sí	 Sí	 Sí
<u>AWS::EC2::IpPoolRouteTableAssociation</u>	 Sí	 Sí	 No
<u>AWS::EC2::KeyPair</u>	 Sí	 Sí	 Sí
<u>AWS::EC2::LaunchTemplate</u>	 Sí	 Sí	 No
<u>AWS::EC2::LocalGatewayRoute</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::EC2::LocalGatewayRouteTable</u>	 Sí	 Sí	 Sí
<u>AWS::EC2::LocalGatewayRouteTableVPCAssociation</u>	 Sí	 Sí	 Sí
<u>AWS::EC2::LocalGatewayRouteTableVirtualInterfaceGroupAssociation</u>	 Sí	 Sí	 Sí
<u>AWS::EC2::LocalGatewayVirtualInterface</u>	 Sí	 Sí	 No
<u>AWS::EC2::LocalGatewayVirtualInterfaceGroup</u>	 Sí	 Sí	 No
<u>AWS::EC2::NatGateway</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::EC2::NetworkAcl</u>	 Sí	 Sí	 Sí
<u>AWS::EC2::NetworkInsightsAccessScope</u>	 Sí	 Sí	 No
<u>AWS::EC2::NetworkInsightsAccessScopeAnalysis</u>	 Sí	 Sí	 Sí
<u>AWS::EC2::NetworkInsightsAnalysis</u>	 Sí	 Sí	 No
<u>AWS::EC2::NetworkInsightsPath</u>	 Sí	 Sí	 Sí
<u>AWS::EC2::NetworkInterface</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::EC2::NetworkInterfaceAttachment</u>	 Sí	 Sí	 Sí
<u>AWS::EC2::NetworkPerformanceMetricSubscription</u>	 Sí	 Sí	 Sí
<u>AWS::EC2::PlacementGroup</u>	 Sí	 Sí	 Sí
<u>AWS::EC2::PrefixList</u>	 Sí	 Sí	 No
<u>AWS::EC2::Route</u>	 Sí	 Sí	 Sí
<u>AWS::EC2::RouteServer</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::EC2::RouteServerAssociation</u>	 Sí	 Sí	 No
<u>AWS::EC2::RouteServerEndpoint</u>	 Sí	 Sí	 No
<u>AWS::EC2::RouteServerPeer</u>	 Sí	 Sí	 No
<u>AWS::EC2::RouteServerPropagation</u>	 Sí	 Sí	 No
<u>AWS::EC2::RouteTable</u>	 Sí	 Sí	 Sí
<u>AWS::EC2::SecurityGroup</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::EC2::SecurityGroupEgress</u>	 Sí	 Sí	 No
<u>AWS::EC2::SecurityGroupIngress</u>	 Sí	 Sí	 No
<u>AWS::EC2::SecurityGroupVpcAssociation</u>	 Sí	 Sí	 No
<u>AWS::EC2::SnapshotBlockPublicAccess</u>	 Sí	 Sí	 No
<u>AWS::EC2::SpotFleet</u>	 Sí	 Sí	 No
<u>AWS::EC2::Subnet</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::EC2::SubnetCidrBlock</u>	 Sí	 Sí	 Sí
<u>AWS::EC2::SubnetNetworkAclAssociation</u>	 Sí	 Sí	 Sí
<u>AWS::EC2::SubnetRouteTableAssociation</u>	 Sí	 No	 Sí
<u>AWS::EC2::TrafficMirrorFilter</u>	 Sí	 Sí	 No
<u>AWS::EC2::TrafficMirrorFilterRule</u>	 Sí	 Sí	 No
<u>AWS::EC2::TrafficMirrorSession</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::EC2::TrafficMirrorTarget</u>	 Sí	 Sí	 No
<u>AWS::EC2::TransitGateway</u>	 Sí	 Sí	 Sí
<u>AWS::EC2::TransitGatewayAttachment</u>	 Sí	 Sí	 Sí
<u>AWS::EC2::TransitGatewayConnect</u>	 Sí	 Sí	 Sí
<u>AWS::EC2::TransitGatewayConnectPeer</u>	 Sí	 Sí	 No
<u>AWS::EC2::TransitGatewayMeteringPolicy</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::EC2::TransitGatewayMeteringPolicyEntry</u>	 Sí	 Sí	 No
<u>AWS::EC2::TransitGatewayMulticastDomain</u>	 Sí	 Sí	 Sí
<u>AWS::EC2::TransitGatewayMulticastDomainAssociation</u>	 Sí	 Sí	 Sí
<u>AWS::EC2::TransitGatewayMulticastGroupMember</u>	 Sí	 Sí	 Sí
<u>AWS::EC2::TransitGatewayMulticastGroupSource</u>	 Sí	 Sí	 Sí
<u>AWS::EC2::TransitGatewayPeeringAttachment</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::EC2::TransitGatewayRoute</u>	 Sí	 Sí	 No
<u>AWS::EC2::TransitGatewayRouteTable</u>	 Sí	 Sí	 Sí
<u>AWS::EC2::TransitGatewayRouteTableAssociation</u>	 Sí	 Sí	 No
<u>AWS::EC2::TransitGatewayRouteTablePropagation</u>	 Sí	 Sí	 No
<u>AWS::EC2::TransitGatewayVpcAttachment</u>	 Sí	 Sí	 Sí
<u>AWS::EC2::VPC</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::EC2::VPCLockPublicAccessExclusion</u>	 Sí	 Sí	 No
<u>AWS::EC2::VPCLockPublicAccessOptions</u>	 Sí	 Sí	 No
<u>AWS::EC2::VPCCIDRBlock</u>	 Sí	 Sí	 No
<u>AWS::EC2::VPCDHCPOptionsAssociation</u>	 Sí	 Sí	 Sí
<u>AWS::EC2::VPCEncryptionControl</u>	 Sí	 Sí	 No
<u>AWS::EC2::VPCEndpoint</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::EC2::VPCEndpointConnectionNotification</u>	 Sí	 Sí	 Sí
<u>AWS::EC2::VPCEndpointService</u>	 Sí	 Sí	 Sí
<u>AWS::EC2::VPCEndpointServicePermissions</u>	 Sí	 Sí	 No
<u>AWS::EC2::VPCGatewayAttachment</u>	 Sí	 Sí	 No
<u>AWS::EC2::VPCPeeringConnection</u>	 Sí	 Sí	 Sí
<u>AWS::EC2::VPNConcentrator</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::EC2::VPNConnection</u>	 Sí	 Sí	 Sí
<u>AWS::EC2::VPNConnectionRoute</u>	 Sí	 Sí	 Sí
<u>AWS::EC2::VPNGateway</u>	 Sí	 Sí	 Sí
<u>AWS::EC2::VerifiedAccessEndpoint</u>	 Sí	 Sí	 Sí
<u>AWS::EC2::VerifiedAccessGroup</u>	 Sí	 Sí	 Sí
<u>AWS::EC2::VerifiedAccessInstance</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::EC2::VerifiedAccessTrustProvider</u>	 Sí	 Sí	 Sí
<u>AWS::EC2::Volume</u>	 Sí	 Sí	 Sí
<u>AWS::EC2::VolumeAttachment</u>	 Sí	 Sí	 Sí
<u>AWS::ECR::PublicRepository</u>	 Sí	 Sí	 Sí
<u>AWS::ECR::PullThroughCacheRule</u>	 Sí	 Sí	 Sí
<u>AWS::ECR::PullTimeUpdateExclusion</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::ECR::RegistryPolicy</u>	 Sí	 Sí	 Sí
<u>AWS::ECR::RegistryScanningConfiguration</u>	 Sí	 Sí	 No
<u>AWS::ECR::ReplicationConfiguration</u>	 Sí	 Sí	 Sí
<u>AWS::ECR::Repository</u>	 Sí	 Sí	 Sí
<u>AWS::ECR::RepositoryCreationTemplate</u>	 Sí	 Sí	 No
<u>AWS::ECR::SigningConfiguration</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::ECS::CapacityProvider</u>	 Sí	 Sí	 No
<u>AWS::ECS::Cluster</u>	 Sí	 No	 Sí
<u>AWS::ECS::ClusterCapacityProviderAssociations</u>	 Sí	 No	 Sí
<u>AWS::ECS::ExpressGatewayService</u>	 Sí	 Sí	 No
<u>AWS::ECS::PrimaryTaskSet</u>	 Sí	 Sí	 No
<u>AWS::ECS::Service</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::ECS::TaskDefinition</u>	 Sí	 Sí	 Sí
<u>AWS::ECS::TaskSet</u>	 Sí	 Sí	 No
<u>AWS::EFS::AccessPoint</u>	 Sí	 Sí	 Sí
<u>AWS::EFS::FileSystem</u>	 Sí	 Sí	 Sí
<u>AWS::EFS::MountTarget</u>	 Sí	 Sí	 Sí
<u>AWS::EKS::AccessEntry</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
AWS::EKS::Addon	 Sí	 Sí	 Sí
AWS::EKS::Capability	 Sí	 Sí	 No
AWS::EKS::Cluster	 Sí	 No	 Sí
AWS::EKS::FargateProfile	 Sí	 Sí	 Sí
AWS::EKS::IdentityProviderConfig	 Sí	 Sí	 Sí
AWS::EKS::Nodegroup	 Sí	 No	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::EKS::PodIdentityAssociation</u>	 Sí	 Sí	 No
<u>AWS::EMR::SecurityConfiguration</u>	 Sí	 Sí	 Sí
<u>AWS::EMR::Step</u>	 Sí	 No	 No
<u>AWS::EMR::Studio</u>	 Sí	 Sí	 Sí
<u>AWS::EMR::StudioSessionMapping</u>	 Sí	 Sí	 Sí
<u>AWS::EMR::WALWorkspace</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::EMRContainers::VirtualCluster</u>	 Sí	 Sí	 No
<u>AWS::EMRServerless::Application</u>	 Sí	 Sí	 Sí
<u>AWS::EVS::Environment</u>	 Sí	 Sí	 No
<u>AWS::ElastiCache::GlobalReplicationGroup</u>	 Sí	 Sí	 No
<u>AWS::ElastiCache::ParameterGroup</u>	 Sí	 Sí	 No
<u>AWS::ElastiCache::ServerlessCache</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::ElastiCache::SubnetGroup</u>	 Sí	 Sí	 Sí
<u>AWS::ElastiCache::User</u>	 Sí	 Sí	 Sí
<u>AWS::ElastiCache::UserGroup</u>	 Sí	 Sí	 Sí
<u>AWS::ElasticBeanstalk::Application</u>	 Sí	 Sí	 Sí
<u>AWS::ElasticBeanstalk::ApplicationVersion</u>	 Sí	 Sí	 Sí
<u>AWS::ElasticBeanstalk::ConfigurationTemplate</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::ElasticBeanstalk::Environment</u>	 Sí	 No	 No
<u>AWS::ElasticLoadBalancingV2::Listener</u>	 Sí	 Sí	 Sí
<u>AWS::ElasticLoadBalancingV2::ListenerRule</u>	 Sí	 No	 Sí
<u>AWS::ElasticLoadBalancingV2::LoadBalancer</u>	 Sí	 Sí	 Sí
<u>AWS::ElasticLoadBalancingV2::TargetGroup</u>	 Sí	 Sí	 Sí
<u>AWS::ElasticLoadBalancingV2::TrustStore</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::ElasticLoadBalancingV2::TrustStoreRevocation</u>	 Sí	 Sí	 No
<u>AWS::EntityResolution::IdMappingWorkflow</u>	 Sí	 Sí	 Sí
<u>AWS::EntityResolution::IdNamespace</u>	 Sí	 Sí	 No
<u>AWS::EntityResolution::MatchingWorkflow</u>	 Sí	 Sí	 Sí
<u>AWS::EntityResolution::PolicyStatement</u>	 Sí	 Sí	 No
<u>AWS::EntityResolution::SchemaMapping</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::EventSchemas::Discoverer</u>	 Sí	 Sí	 No
<u>AWS::EventSchemas::Registry</u>	 Sí	 Sí	 No
<u>AWS::EventSchemas::RegistryPolicy</u>	 Sí	 Sí	 No
<u>AWS::EventSchemas::Schema</u>	 Sí	 Sí	 No
<u>AWS::Events::ApiDestination</u>	 Sí	 Sí	 Sí
<u>AWS::Events::Archive</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::Events::Connection</u>	 Sí	 Sí	 No
<u>AWS::Events::Endpoint</u>	 Sí	 Sí	 Sí
<u>AWS::Events::EventBus</u>	 Sí	 Sí	 Sí
<u>AWS::Events::EventBusPolicy</u>	 Sí	 Sí	 No
<u>AWS::Events::Rule</u>	 Sí	 Sí	 Sí
<u>AWS::Evidently::Experiment</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::Evidently::Feature</u>	 Sí	 Sí	 No
<u>AWS::Evidently::Launch</u>	 Sí	 Sí	 No
<u>AWS::Evidently::Project</u>	 Sí	 Sí	 No
<u>AWS::Evidently::Segment</u>	 Sí	 Sí	 No
<u>AWS::FIS::ExperimentTemplate</u>	 Sí	 Sí	 Sí
<u>AWS::FIS::TargetAccountConfiguration</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::FMS::NotificationChannel</u>	 Sí	 Sí	 No
<u>AWS::FMS::Policy</u>	 Sí	 No	 No
<u>AWS::FMS::ResourceSet</u>	 Sí	 Sí	 No
<u>AWS::FSx::DataRepositoryAssociation</u>	 Sí	 Sí	 Sí
<u>AWS::FSx::S3AccessPointAttachment</u>	 Sí	 Sí	 No
<u>AWS::FinSpace::Environment</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::Forecast::Dataset</u>	 Sí	 Sí	 No
<u>AWS::Forecast::DatasetGroup</u>	 Sí	 Sí	 No
<u>AWS::FraudDetector::Detector</u>	 Sí	 Sí	 Sí
<u>AWS::FraudDetector::EntityType</u>	 Sí	 Sí	 Sí
<u>AWS::FraudDetector::EventType</u>	 Sí	 Sí	 Sí
<u>AWS::FraudDetector::Label</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::FraudDetector::List</u>	 Sí	 Sí	 Sí
<u>AWS::FraudDetector::Outcome</u>	 Sí	 Sí	 Sí
<u>AWS::FraudDetector::Variable</u>	 Sí	 Sí	 Sí
<u>AWS::GameLift::Alias</u>	 Sí	 Sí	 Sí
<u>AWS::GameLift::Build</u>	 Sí	 Sí	 Sí
<u>AWS::GameLift::ContainerFleet</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::GameLift::ContainerGroupDefinition</u>	 Sí	 Sí	 No
<u>AWS::GameLift::Fleet</u>	 Sí	 Sí	 Sí
<u>AWS::GameLift::GameServerGroup</u>	 Sí	 Sí	 No
<u>AWS::GameLift::GameSessionQueue</u>	 Sí	 Sí	 No
<u>AWS::GameLift::Location</u>	 Sí	 Sí	 No
<u>AWS::GameLift::MatchmakingConfiguration</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::GameLift::MatchmakingRuleSet</u>	 Sí	 Sí	 No
<u>AWS::GameLift::Script</u>	 Sí	 Sí	 No
<u>AWS::GameLiftStreams::Application</u>	 Sí	 Sí	 No
<u>AWS::GameLiftStreams::StreamGroup</u>	 Sí	 Sí	 No
<u>AWS::GlobalAccelerator::Accelerator</u>	 Sí	 Sí	 Sí
<u>AWS::GlobalAccelerator::CrossAccount Attachment</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::GlobalAccelerator::EndpointGroup</u>	 Sí	 Sí	 Sí
<u>AWS::GlobalAccelerator::Listener</u>	 Sí	 Sí	 Sí
<u>AWS::Glue::Crawler</u>	 Sí	 Sí	 No
<u>AWS::Glue::Database</u>	 Sí	 Sí	 No
<u>AWS::Glue::IdentityCenterConfiguration</u>	 Sí	 Sí	 No
<u>AWS::Glue::Integration</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::Glue::IntegrationResourceProperty</u>	 Sí	 Sí	 No
<u>AWS::Glue::Job</u>	 Sí	 Sí	 No
<u>AWS::Glue::Registry</u>	 Sí	 Sí	 Sí
<u>AWS::Glue::Schema</u>	 Sí	 Sí	 Sí
<u>AWS::Glue::SchemaVersion</u>	 Sí	 Sí	 Sí
<u>AWS::Glue::SchemaVersionMetadata</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::Glue::Trigger</u>	 Sí	 Sí	 No
<u>AWS::Glue::UsageProfile</u>	 Sí	 Sí	 No
<u>AWS::Grafana::Workspace</u>	 Sí	 Sí	 Sí
<u>AWS::GreengrassV2::ComponentVersion</u>	 Sí	 Sí	 No
<u>AWS::GreengrassV2::Deployment</u>	 Sí	 Sí	 Sí
<u>AWS::GroundStation::Config</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::GroundStation::DataflowEndpointGroup</u>	 Sí	 Sí	 Sí
<u>AWS::GroundStation::DataflowEndpointGroupV2</u>	 Sí	 Sí	 No
<u>AWS::GroundStation::MissionProfile</u>	 Sí	 Sí	 Sí
<u>AWS::GuardDuty::Detector</u>	 Sí	 Sí	 Sí
<u>AWS::GuardDuty::Filter</u>	 Sí	 Sí	 No
<u>AWS::GuardDuty::IPSet</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::GuardDuty::MalwareProtectionPlan</u>	 Sí	 Sí	 No
<u>AWS::GuardDuty::Master</u>	 Sí	 Sí	 No
<u>AWS::GuardDuty::Member</u>	 Sí	 Sí	 No
<u>AWS::GuardDuty::PublishingDestination</u>	 Sí	 Sí	 No
<u>AWS::GuardDuty::ThreatEntitySet</u>	 Sí	 Sí	 No
<u>AWS::GuardDuty::ThreatIntelSet</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::GuardDuty::TrustedEntitySet</u>	 Sí	 Sí	 No
<u>AWS::HealthImaging::Datastore</u>	 Sí	 Sí	 Sí
<u>AWS::HealthLake::FHIRDatastore</u>	 Sí	 Sí	 Sí
<u>AWS::IAM::Group</u>	 Sí	 Sí	 Sí
<u>AWS::IAM::GroupPolicy</u>	 Sí	 Sí	 No
<u>AWS::IAM::InstanceProfile</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::IAM::ManagedPolicy</u>	 Sí	 Sí	 Sí
<u>AWS::IAM::OIDCProvider</u>	 Sí	 Sí	 Sí
<u>AWS::IAM::Role</u>	 Sí	 Sí	 Sí
<u>AWS::IAM::RolePolicy</u>	 Sí	 Sí	 No
<u>AWS::IAM::SAMLProvider</u>	 Sí	 Sí	 Sí
<u>AWS::IAM::ServerCertificate</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::IAM::ServiceLinkedRole</u>	 Sí	 Sí	 No
<u>AWS::IAM::User</u>	 Sí	 Sí	 Sí
<u>AWS::IAM::UserPolicy</u>	 Sí	 Sí	 No
<u>AWS::IAM::VirtualMFADevice</u>	 Sí	 Sí	 No
<u>AWS::IVS::Channel</u>	 Sí	 Sí	 Sí
<u>AWS::IVS::EncoderConfiguration</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::IVS::IngestConfiguration</u>	 Sí	 Sí	 No
<u>AWS::IVS::PlaybackKeyPair</u>	 Sí	 Sí	 Sí
<u>AWS::IVS::PlaybackRestrictionPolicy</u>	 Sí	 Sí	 No
<u>AWS::IVS::PublicKey</u>	 Sí	 Sí	 No
<u>AWS::IVS::RecordingConfiguration</u>	 Sí	 Sí	 Sí
<u>AWS::IVS::Stage</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::IVS::StorageConfiguration</u>	 Sí	 Sí	 No
<u>AWS::IVS::StreamKey</u>	 Sí	 Sí	 No
<u>AWS::IVSChat::LoggingConfiguration</u>	 Sí	 Sí	 Sí
<u>AWS::IVSChat::Room</u>	 Sí	 Sí	 Sí
<u>AWS::IdentityStore::Group</u>	 Sí	 Sí	 Sí
<u>AWS::IdentityStore::GroupMembership</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::ImageBuilder::Component</u>	 Sí	 Sí	 Sí
<u>AWS::ImageBuilder::ContainerRecipe</u>	 Sí	 Sí	 No
<u>AWS::ImageBuilder::DistributionConfiguration</u>	 Sí	 Sí	 Sí
<u>AWS::ImageBuilder::Image</u>	 Sí	 Sí	 Sí
<u>AWS::ImageBuilder::ImagePipeline</u>	 Sí	 Sí	 Sí
<u>AWS::ImageBuilder::ImageRecipe</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::ImageBuilder::InfrastructureConfiguration</u>	 Sí	 Sí	 Sí
<u>AWS::ImageBuilder::LifecyclePolicy</u>	 Sí	 Sí	 No
<u>AWS::ImageBuilder::Workflow</u>	 Sí	 Sí	 No
<u>AWS::Inspector::AssessmentTarget</u>	 Sí	 Sí	 Sí
<u>AWS::Inspector::AssessmentTemplate</u>	 Sí	 Sí	 Sí
<u>AWS::Inspector::ResourceGroup</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::InspectorV2::CisScanConfiguration</u>	 Sí	 Sí	 No
<u>AWS::InspectorV2::CodeSecurityIntegration</u>	 Sí	 Sí	 No
<u>AWS::InspectorV2::CodeSecurityScanConfiguration</u>	 Sí	 Sí	 No
<u>AWS::InspectorV2::Filter</u>	 Sí	 Sí	 No
<u>AWS::InternetMonitor::Monitor</u>	 Sí	 Sí	 Sí
<u>AWS::Invoicing::InvoiceUnit</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::IoT::AccountAuditConfiguration</u>	 Sí	 Sí	 Sí
<u>AWS::IoT::Authorizer</u>	 Sí	 Sí	 Sí
<u>AWS::IoT::BillingGroup</u>	 Sí	 Sí	 Sí
<u>AWS::IoT::CACertificate</u>	 Sí	 Sí	 Sí
<u>AWS::IoT::Certificate</u>	 Sí	 Sí	 Sí
<u>AWS::IoT::CertificateProvider</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::IoT::Command</u>	 Sí	 Sí	 No
<u>AWS::IoT::CustomMetric</u>	 Sí	 Sí	 Sí
<u>AWS::IoT::Dimension</u>	 Sí	 Sí	 Sí
<u>AWS::IoT::DomainConfiguration</u>	 Sí	 Sí	 No
<u>AWS::IoT::EncryptionConfiguration</u>	 Sí	 Sí	 No
<u>AWS::IoT::FleetMetric</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::IoT::JobTemplate</u>	 Sí	 Sí	 No
<u>AWS::IoT::Logging</u>	 Sí	 Sí	 Sí
<u>AWS::IoT::MitigationAction</u>	 Sí	 Sí	 Sí
<u>AWS::IoT::Policy</u>	 Sí	 Sí	 Sí
<u>AWS::IoT::ProvisioningTemplate</u>	 Sí	 Sí	 Sí
<u>AWS::IoT::ResourceSpecificLogging</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::IoT::RoleAlias</u>	 Sí	 Sí	 Sí
<u>AWS::IoT::ScheduledAudit</u>	 Sí	 Sí	 Sí
<u>AWS::IoT::SecurityProfile</u>	 Sí	 Sí	 Sí
<u>AWS::IoT::SoftwarePackage</u>	 Sí	 Sí	 Sí
<u>AWS::IoT::SoftwarePackageVersion</u>	 Sí	 Sí	 No
<u>AWS::IoT::Thing</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::IoT::ThingGroup</u>	 Sí	 Sí	 Sí
<u>AWS::IoT::ThingType</u>	 Sí	 Sí	 No
<u>AWS::IoT::TopicRule</u>	 Sí	 Sí	 Sí
<u>AWS::IoT::TopicRuleDestination</u>	 Sí	 Sí	 Sí
<u>AWS::IoTAnalytics::Channel</u>	 Sí	 Sí	 Sí
<u>AWS::IoTAnalytics::Dataset</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::IoTAnalytics::Datastore</u>	 Sí	 Sí	 Sí
<u>AWS::IoTAnalytics::Pipeline</u>	 Sí	 Sí	 Sí
<u>AWS::IoTCoreDeviceAdvisor::SuiteDefinition</u>	 Sí	 Sí	 Sí
<u>AWS::IoTEvents::AlarmModel</u>	 Sí	 Sí	 Sí
<u>AWS::IoTEvents::DetectorModel</u>	 Sí	 Sí	 Sí
<u>AWS::IoTEvents::Input</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::IoT FleetWise::Campaign</u>	 Sí	 Sí	 Sí
<u>AWS::IoT FleetWise::DecoderManifest</u>	 Sí	 Sí	 Sí
<u>AWS::IoT FleetWise::Fleet</u>	 Sí	 Sí	 Sí
<u>AWS::IoT FleetWise::ModelManifest</u>	 Sí	 Sí	 Sí
<u>AWS::IoT FleetWise::SignalCatalog</u>	 Sí	 Sí	 Sí
<u>AWS::IoT FleetWise::StateTemplate</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::IoT FleetWise::Vehicle</u>	 Sí	 Sí	 Sí
<u>AWS::IoT SiteWise::AccessPolicy</u>	 Sí	 Sí	 No
<u>AWS::IoT SiteWise::Asset</u>	 Sí	 Sí	 Sí
<u>AWS::IoT SiteWise::AssetModel</u>	 Sí	 Sí	 Sí
<u>AWS::IoT SiteWise::ComputationModel</u>	 Sí	 Sí	 No
<u>AWS::IoT SiteWise::Dashboard</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::IoTSiteWise::Dataset</u>	 Sí	 Sí	 No
<u>AWS::IoTSiteWise::Gateway</u>	 Sí	 Sí	 Sí
<u>AWS::IoTSiteWise::Portal</u>	 Sí	 Sí	 No
<u>AWS::IoTSiteWise::Project</u>	 Sí	 Sí	 No
<u>AWS::IoTtwinMaker::ComponentType</u>	 Sí	 Sí	 Sí
<u>AWS::IoTtwinMaker::Entity</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::IoT TwinMaker::Scene</u>	 Sí	 Sí	 Sí
<u>AWS::IoT TwinMaker::SyncJob</u>	 Sí	 Sí	 Sí
<u>AWS::IoT TwinMaker::Workspace</u>	 Sí	 Sí	 Sí
<u>AWS::IoT Wireless::Destination</u>	 Sí	 Sí	 Sí
<u>AWS::IoT Wireless::DeviceProfile</u>	 Sí	 Sí	 Sí
<u>AWS::IoT Wireless::FleetTask</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::IoTWireless::MulticastGroup</u>	 Sí	 Sí	 Sí
<u>AWS::IoTWireless::NetworkAnalyzerConfiguration</u>	 Sí	 Sí	 Sí
<u>AWS::IoTWireless::PartnerAccount</u>	 Sí	 Sí	 Sí
<u>AWS::IoTWireless::ServiceProfile</u>	 Sí	 Sí	 Sí
<u>AWS::IoTWireless::TaskDefinition</u>	 Sí	 Sí	 Sí
<u>AWS::IoTWireless::WirelessDevice</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::IoTWireless::WirelessDeviceImportTask</u>	 Sí	 Sí	 Sí
<u>AWS::IoTWireless::WirelessGateway</u>	 Sí	 Sí	 Sí
<u>AWS::KMS::Alias</u>	 Sí	 No	 Sí
<u>AWS::KMS::Key</u>	 Sí	 No	 Sí
<u>AWS::KMS::ReplicaKey</u>	 Sí	 No	 Sí
<u>AWS::KafkaConnect::Connector</u>	 Sí	 No	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::KafkaConnect::CustomPlugin</u>	 Sí	 Sí	 No
<u>AWS::KafkaConnect::WorkerConfiguration</u>	 Sí	 Sí	 No
<u>AWS::Kendra::DataSource</u>	 Sí	 Sí	 Sí
<u>AWS::Kendra::FAQ</u>	 Sí	 Sí	 Sí
<u>AWS::Kendra::Index</u>	 Sí	 Sí	 Sí
<u>AWS::KendraRanking::ExecutionPlan</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::Kinesis::ResourcePolicy</u>	 Sí	 Sí	 No
<u>AWS::Kinesis::Stream</u>	 Sí	 Sí	 Sí
<u>AWS::Kinesis::StreamConsumer</u>	 Sí	 Sí	 No
<u>AWS::KinesisAnalyticsV2::Application</u>	 Sí	 Sí	 No
<u>AWS::KinesisFirehose::DeliveryStream</u>	 Sí	 No	 No
<u>AWS::KinesisVideo::SignalingChannel</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::KinesisVideo::Stream</u>	 Sí	 Sí	 No
<u>AWS::LakeFormation::DataCellsFilter</u>	 Sí	 Sí	 No
<u>AWS::LakeFormation::PrincipalPermissions</u>	 Sí	 Sí	 No
<u>AWS::LakeFormation::Tag</u>	 Sí	 Sí	 No
<u>AWS::LakeFormation::TagAssociation</u>	 Sí	 Sí	 No
<u>AWS::Lambda::Alias</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::Lambda::CapacityProvider</u>	 Sí	 Sí	 No
<u>AWS::Lambda::CodeSigningConfig</u>	 Sí	 Sí	 Sí
<u>AWS::Lambda::EventInvokeConfig</u>	 Sí	 Sí	 No
<u>AWS::Lambda::EventSourceMapping</u>	 Sí	 Sí	 Sí
<u>AWS::Lambda::Function</u>	 Sí	 Sí	 Sí
<u>AWS::Lambda::LayerVersion</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::Lambda::LayerVersionPermission</u>	 Sí	 Sí	 Sí
<u>AWS::Lambda::Permission</u>	 Sí	 Sí	 Sí
<u>AWS::Lambda::Url</u>	 Sí	 Sí	 Sí
<u>AWS::Lambda::Version</u>	 Sí	 Sí	 Sí
<u>AWS::LaunchWizard::Deployment</u>	 Sí	 Sí	 No
<u>AWS::Lex::Bot</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::Lex::BotAlias</u>	 Sí	 Sí	 No
<u>AWS::Lex::BotVersion</u>	 Sí	 Sí	 No
<u>AWS::Lex::ResourcePolicy</u>	 Sí	 Sí	 No
<u>AWS::LicenseManager::Grant</u>	 Sí	 Sí	 No
<u>AWS::LicenseManager::License</u>	 Sí	 Sí	 No
<u>AWS::Lightsail::Alarm</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
AWS::Lightsail::Bucket	 Sí	 Sí	 Sí
AWS::Lightsail::Certificate	 Sí	 Sí	 Sí
AWS::Lightsail::Container	 Sí	 Sí	 Sí
AWS::Lightsail::Database	 Sí	 Sí	 No
AWS::Lightsail::Disk	 Sí	 Sí	 Sí
AWS::Lightsail::DiskSnapshot	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::Lightsail::Distribution</u>	 Sí	 Sí	 Sí
<u>AWS::Lightsail::Domain</u>	 Sí	 Sí	 No
<u>AWS::Lightsail::Instance</u>	 Sí	 Sí	 Sí
<u>AWS::Lightsail::InstanceSnapshot</u>	 Sí	 Sí	 No
<u>AWS::Lightsail::LoadBalancer</u>	 Sí	 Sí	 Sí
<u>AWS::Lightsail::LoadBalancerTlsCertificate</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
AWS::Lightsail::StaticIp	 Sí	 Sí	 Sí
AWS::Location::APIKey	 Sí	 Sí	 No
AWS::Location::GeofenceCollection	 Sí	 Sí	 Sí
AWS::Location::Map	 Sí	 Sí	 Sí
AWS::Location::PlaceIndex	 Sí	 Sí	 Sí
AWS::Location::RouteCalculator	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::Location::Tracker</u>	 Sí	 Sí	 Sí
<u>AWS::Location::TrackerConsumer</u>	 Sí	 Sí	 Sí
<u>AWS::Logs::AccountPolicy</u>	 Sí	 Sí	 Sí
<u>AWS::Logs::Delivery</u>	 Sí	 Sí	 No
<u>AWS::Logs::DeliveryDestination</u>	 Sí	 Sí	 No
<u>AWS::Logs::DeliverySource</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::Logs::Destination</u>	 Sí	 Sí	 Sí
<u>AWS::Logs::Integration</u>	 Sí	 Sí	 No
<u>AWS::Logs::LogAnomalyDetector</u>	 Sí	 Sí	 No
<u>AWS::Logs::LogGroup</u>	 Sí	 Sí	 Sí
<u>AWS::Logs::LogStream</u>	 Sí	 Sí	 Sí
<u>AWS::Logs::MetricFilter</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::Logs::QueryDefinition</u>	 Sí	 Sí	 No
<u>AWS::Logs::ResourcePolicy</u>	 Sí	 Sí	 Sí
<u>AWS::Logs::SubscriptionFilter</u>	 Sí	 Sí	 Sí
<u>AWS::Logs::Transformer</u>	 Sí	 Sí	 No
<u>AWS::LookoutEquipment::InferenceScheduler</u>	 Sí	 Sí	 No
<u>AWS::LookoutVision::Project</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::M2::Application</u>	 Sí	 Sí	 No
<u>AWS::M2::Deployment</u>	 Sí	 Sí	 No
<u>AWS::M2::Environment</u>	 Sí	 Sí	 Sí
<u>AWS::MPA::ApprovalTeam</u>	 Sí	 Sí	 No
<u>AWS::MPA::IdentitySource</u>	 Sí	 Sí	 No
<u>AWS::MSK::BatchScramSecret</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::MSK::Cluster</u>	 Sí	 Sí	 Sí
<u>AWS::MSK::ClusterPolicy</u>	 Sí	 Sí	 Sí
<u>AWS::MSK::Configuration</u>	 Sí	 Sí	 Sí
<u>AWS::MSK::Replicator</u>	 Sí	 Sí	 Sí
<u>AWS::MSK::ServerlessCluster</u>	 Sí	 Sí	 Sí
<u>AWS::MSK::VpcConnection</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::MWA::Environment</u>	 Sí	 Sí	 No
<u>AWS::Macie::AllowList</u>	 Sí	 Sí	 No
<u>AWS::Macie::CustomDataIdentifier</u>	 Sí	 Sí	 No
<u>AWS::Macie::FindingsFilter</u>	 Sí	 Sí	 No
<u>AWS::Macie::Session</u>	 Sí	 Sí	 Sí
<u>AWS::ManagedBlockchain::Accessor</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::MediaConnect::Bridge</u>	 Sí	 Sí	 Sí
<u>AWS::MediaConnect::BridgeOutput</u>	 Sí	 Sí	 No
<u>AWS::MediaConnect::BridgeSource</u>	 Sí	 Sí	 No
<u>AWS::MediaConnect::Flow</u>	 Sí	 Sí	 No
<u>AWS::MediaConnect::FlowEntitlement</u>	 Sí	 Sí	 No
<u>AWS::MediaConnect::FlowOutput</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::MediaConnect::FlowSource</u>	 Sí	 Sí	 No
<u>AWS::MediaConnect::FlowVpcInterface</u>	 Sí	 Sí	 Sí
<u>AWS::MediaConnect::Gateway</u>	 Sí	 Sí	 Sí
<u>AWS::MediaConnect::RouterInput</u>	 Sí	 Sí	 No
<u>AWS::MediaConnect::RouterNetworkInterface</u>	 Sí	 Sí	 No
<u>AWS::MediaConnect::RouterOutput</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::MediaLive::ChannelPlacementGroup</u>	 Sí	 Sí	 No
<u>AWS::MediaLive::CloudWatchAlarmTemplate</u>	 Sí	 Sí	 No
<u>AWS::MediaLive::CloudWatchAlarmTemplateGroup</u>	 Sí	 Sí	 No
<u>AWS::MediaLive::Cluster</u>	 Sí	 Sí	 No
<u>AWS::MediaLive::EventBridgeRuleTemplate</u>	 Sí	 Sí	 No
<u>AWS::MediaLive::EventBridgeRuleTemplateGroup</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::MediaLive::Multiplex</u>	 Sí	 Sí	 No
<u>AWS::MediaLive::Multiplexprogram</u>	 Sí	 Sí	 No
<u>AWS::MediaLive::Network</u>	 Sí	 Sí	 No
<u>AWS::MediaLive::SdiSource</u>	 Sí	 Sí	 No
<u>AWS::MediaLive::SignalMap</u>	 Sí	 Sí	 No
<u>AWS::MediaPackage::Asset</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::MediaPackage::Channel</u>	 Sí	 Sí	 Sí
<u>AWS::MediaPackage::OriginEndpoint</u>	 Sí	 Sí	 Sí
<u>AWS::MediaPackage::PackagingConfiguration</u>	 Sí	 Sí	 Sí
<u>AWS::MediaPackage::PackagingGroup</u>	 Sí	 Sí	 Sí
<u>AWS::MediaPackageV2::Channel</u>	 Sí	 Sí	 Sí
<u>AWS::MediaPackageV2::ChannelGroup</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::MediaPackageV2::ChannelPolicy</u>	 Sí	 Sí	 No
<u>AWS::MediaPackageV2::OriginEndpoint</u>	 Sí	 Sí	 Sí
<u>AWS::MediaPackageV2::OriginEndpointPolicy</u>	 Sí	 Sí	 No
<u>AWS::MediaTailor::Channel</u>	 Sí	 Sí	 No
<u>AWS::MediaTailor::ChannelPolicy</u>	 Sí	 Sí	 No
<u>AWS::MediaTailor::LiveSource</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::MediaTailor::PlaybackConfiguration</u>	 Sí	 Sí	 No
<u>AWS::MediaTailor::SourceLocation</u>	 Sí	 Sí	 Sí
<u>AWS::MediaTailor::VodSource</u>	 Sí	 Sí	 Sí
<u>AWS::MemoryDB::ACL</u>	 Sí	 Sí	 Sí
<u>AWS::MemoryDB::Cluster</u>	 Sí	 Sí	 Sí
<u>AWS::MemoryDB::MultiRegionCluster</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::MemoryDB::ParameterGroup</u>	 Sí	 Sí	 Sí
<u>AWS::MemoryDB::SubnetGroup</u>	 Sí	 Sí	 Sí
<u>AWS::MemoryDB::User</u>	 Sí	 Sí	 Sí
<u>AWS::Neptune::DBCluster</u>	 Sí	 Sí	 Sí
<u>AWS::Neptune::DBClusterParameterGroup</u>	 Sí	 Sí	 No
<u>AWS::Neptune::DBInstance</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::Neptune::DBParameterGroup</u>	 Sí	 Sí	 No
<u>AWS::Neptune::DBSubnetGroup</u>	 Sí	 Sí	 No
<u>AWS::Neptune::EventSubscription</u>	 Sí	 Sí	 No
<u>AWS::NeptuneGraph::Graph</u>	 Sí	 Sí	 No
<u>AWS::NeptuneGraph::PrivateGraphEndpoint</u>	 Sí	 Sí	 No
<u>AWS::NetworkFirewall::Firewall</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::NetworkFirewall::FirewallPolicy</u>	 Sí	 Sí	 Sí
<u>AWS::NetworkFirewall::LoggingConfiguration</u>	 Sí	 Sí	 No
<u>AWS::NetworkFirewall::RuleGroup</u>	 Sí	 Sí	 Sí
<u>AWS::NetworkFirewall::TLSInspectionConfiguration</u>	 Sí	 Sí	 No
<u>AWS::NetworkFirewall::VpcEndpointAssociation</u>	 Sí	 Sí	 No
<u>AWS::NetworkManager::ConnectAttachment</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::NetworkManager::ConnectPeer</u>	 Sí	 Sí	 Sí
<u>AWS::NetworkManager::CoreNetwork</u>	 Sí	 Sí	 Sí
<u>AWS::NetworkManager::CoreNetworkPrefixListAssociation</u>	 Sí	 Sí	 No
<u>AWS::NetworkManager::CustomerGatewayAssociation</u>	 Sí	 Sí	 Sí
<u>AWS::NetworkManager::Device</u>	 Sí	 Sí	 Sí
<u>AWS::NetworkManager::DirectConnectGatewayAttachment</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::NetworkManager::GlobalNetwork</u>	 Sí	 Sí	 Sí
<u>AWS::NetworkManager::Link</u>	 Sí	 Sí	 Sí
<u>AWS::NetworkManager::LinkAssociation</u>	 Sí	 Sí	 Sí
<u>AWS::NetworkManager::Site</u>	 Sí	 Sí	 Sí
<u>AWS::NetworkManager::SiteToSiteVpnAttachment</u>	 Sí	 Sí	 Sí
<u>AWS::NetworkManager::TransitGatewayPeering</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::NetworkManager::TransitGatewayRegistration</u>	 Sí	 Sí	 Sí
<u>AWS::NetworkManager::TransitGatewayRouteTableAttachment</u>	 Sí	 Sí	 Sí
<u>AWS::NetworkManager::VpcAttachment</u>	 Sí	 Sí	 Sí
<u>AWS::Notifications::ChannelAssociation</u>	 Sí	 Sí	 No
<u>AWS::Notifications::EventRule</u>	 Sí	 Sí	 No
<u>AWS::Notifications::ManagedNotificationAccountContactAssociation</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::Notifications::ManagedNotificationAdditionalChannelAssociation</u>	 Sí	 Sí	 No
<u>AWS::Notifications::NotificationConfiguration</u>	 Sí	 Sí	 No
<u>AWS::Notifications::NotificationHub</u>	 Sí	 Sí	 No
<u>AWS::Notifications::OrganizationalUnitAssociation</u>	 Sí	 Sí	 No
<u>AWS::NotificationsContacts::EmailContact</u>	 Sí	 Sí	 No
<u>AWS::ODB::CloudAutonomousVmCluster</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::ODB::CloudExadataInfrastructure</u>	 Sí	 Sí	 No
<u>AWS::ODB::CloudVmCluster</u>	 Sí	 Sí	 No
<u>AWS::ODB::OdbNetwork</u>	 Sí	 Sí	 No
<u>AWS::ODB::OdbPeeringConnection</u>	 Sí	 Sí	 No
<u>AWS::OSIS::Pipeline</u>	 Sí	 Sí	 Sí
<u>AWS::Oam::Link</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::Oam::Sink</u>	 Sí	 Sí	 Sí
<u>AWS::ObservabilityAdmin::OrganizationCentralizationRule</u>	 Sí	 Sí	 No
<u>AWS::ObservabilityAdmin::OrganizationTelemetryRule</u>	 Sí	 Sí	 No
<u>AWS::ObservabilityAdmin::S3TableIntegration</u>	 Sí	 Sí	 No
<u>AWS::ObservabilityAdmin::TelemetryPipelines</u>	 Sí	 Sí	 No
<u>AWS::ObservabilityAdmin::TelemetryRule</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::Omics::AnnotationStore</u>	 Sí	 Sí	 Sí
<u>AWS::Omics::ReferenceStore</u>	 Sí	 Sí	 Sí
<u>AWS::Omics::RunGroup</u>	 Sí	 Sí	 Sí
<u>AWS::Omics::SequenceStore</u>	 Sí	 Sí	 Sí
<u>AWS::Omics::VariantStore</u>	 Sí	 Sí	 Sí
<u>AWS::Omics::Workflow</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::Omicss::WorkflowVersion</u>	 Sí	 Sí	 No
<u>AWS::OpenSearchServerless::AccessPolicy</u>	 Sí	 Sí	 Sí
<u>AWS::OpenSearchServerless::Collection</u>	 Sí	 Sí	 No
<u>AWS::OpenSearchServerless::CollectionGroup</u>	 Sí	 Sí	 No
<u>AWS::OpenSearchServerless::Index</u>	 Sí	 Sí	 No
<u>AWS::OpenSearchServerless::LifecyclePolicy</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::OpenSearchServerless::SecurityConfig</u>	 Sí	 Sí	 Sí
<u>AWS::OpenSearchServerless::SecurityPolicy</u>	 Sí	 Sí	 Sí
<u>AWS::OpenSearchServerless::VpcEndpoint</u>	 Sí	 Sí	 Sí
<u>AWS::OpenSearchService::Application</u>	 Sí	 Sí	 No
<u>AWS::OpenSearchService::Domain</u>	 Sí	 Sí	 No
<u>AWS::Organizations::Account</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::Organizations::Organization</u>	 Sí	 Sí	 Sí
<u>AWS::Organizations::OrganizationalUnit</u>	 Sí	 Sí	 Sí
<u>AWS::Organizations::Policy</u>	 Sí	 Sí	 Sí
<u>AWS::Organizations::ResourcePolicy</u>	 Sí	 Sí	 Sí
<u>AWS::PCAManagedAD::Connector</u>	 Sí	 Sí	 Sí
<u>AWS::PCAManagedAD::DirectoryRegistration</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::PCAConnectorAD::ServicePrincipalName</u>	 Sí	 Sí	 Sí
<u>AWS::PCAConnectorAD::Template</u>	 Sí	 Sí	 Sí
<u>AWS::PCAConnectorAD::TemplateGroupAccessControlEntry</u>	 Sí	 Sí	 Sí
<u>AWS::PCAConnectorSCEP::Challenge</u>	 Sí	 Sí	 No
<u>AWS::PCAConnectorSCEP::Connector</u>	 Sí	 Sí	 No
<u>AWS::PCS::Cluster</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::PCS::ComputeNodeGroup</u>	 Sí	 Sí	 No
<u>AWS::PCS::Queue</u>	 Sí	 Sí	 No
<u>AWS::Panorama::ApplicationInstance</u>	 Sí	 Sí	 Sí
<u>AWS::Panorama::Package</u>	 Sí	 Sí	 Sí
<u>AWS::Panorama::PackageVersion</u>	 Sí	 Sí	 No
<u>AWS::PaymentCryptography::Alias</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::PaymentCryptography::Key</u>	 Sí	 Sí	 No
<u>AWS::Personalize::Dataset</u>	 Sí	 Sí	 Sí
<u>AWS::Personalize::DatasetGroup</u>	 Sí	 Sí	 Sí
<u>AWS::Personalize::Schema</u>	 Sí	 Sí	 Sí
<u>AWS::Personalize::Solution</u>	 Sí	 Sí	 No
<u>AWS::Pinpoint::InAppTemplate</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::Pipes::Pipe</u>	 Sí	 Sí	 Sí
<u>AWS::Proton::EnvironmentAccountConnection</u>	 Sí	 Sí	 Sí
<u>AWS::Proton::EnvironmentTemplate</u>	 Sí	 Sí	 Sí
<u>AWS::Proton::ServiceTemplate</u>	 Sí	 Sí	 Sí
<u>AWS::QBusiness::Application</u>	 Sí	 Sí	 No
<u>AWS::QBusiness::DataAccessor</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::QBusiness::DataSource</u>	 Sí	 Sí	 No
<u>AWS::QBusiness::Index</u>	 Sí	 Sí	 No
<u>AWS::QBusiness::Permission</u>	 Sí	 Sí	 No
<u>AWS::QBusiness::Plugin</u>	 Sí	 Sí	 No
<u>AWS::QBusiness::Retriever</u>	 Sí	 Sí	 No
<u>AWS::QBusiness::WebExperience</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::QLDB::Stream</u>	 Sí	 Sí	 Sí
<u>AWS::QuickSight::Analysis</u>	 Sí	 Sí	 No
<u>AWS::QuickSight::CustomPermissions</u>	 Sí	 Sí	 No
<u>AWS::QuickSight::Dashboard</u>	 Sí	 Sí	 No
<u>AWS::QuickSight::DataSet</u>	 Sí	 Sí	 Sí
<u>AWS::QuickSight::DataSource</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::QuickSight::Folder</u>	 Sí	 Sí	 No
<u>AWS::QuickSight::RefreshSchedule</u>	 Sí	 Sí	 Sí
<u>AWS::QuickSight::Template</u>	 Sí	 Sí	 No
<u>AWS::QuickSight::Theme</u>	 Sí	 Sí	 No
<u>AWS::QuickSight::Topic</u>	 Sí	 Sí	 Sí
<u>AWS::QuickSight::VPCConnection</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::RAM::Permission</u>	 Sí	 Sí	 No
<u>AWS::RAM::ResourceShare</u>	 Sí	 Sí	 No
<u>AWS::RDS::CustomDBEngineVersion</u>	 Sí	 Sí	 No
<u>AWS::RDS::DBCluster</u>	 Sí	 Sí	 Sí
<u>AWS::RDS::DBClusterParameterGroup</u>	 Sí	 Sí	 No
<u>AWS::RDS::DBInstance</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::RDS::DBParameterGroup</u>	 Sí	 Sí	 No
<u>AWS::RDS::DBProxy</u>	 Sí	 Sí	 Sí
<u>AWS::RDS::DBProxyEndpoint</u>	 Sí	 Sí	 Sí
<u>AWS::RDS::DBProxyTargetGroup</u>	 Sí	 Sí	 Sí
<u>AWS::RDS::DBShardGroup</u>	 Sí	 Sí	 No
<u>AWS::RDS::DBSubnetGroup</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::RDS::EventSubscription</u>	 Sí	 Sí	 Sí
<u>AWS::RDS::GlobalCluster</u>	 Sí	 Sí	 Sí
<u>AWS::RDS::Integration</u>	 Sí	 Sí	 No
<u>AWS::RDS::OptionGroup</u>	 Sí	 Sí	 Sí
<u>AWS::RTBFabric::InboundExternalLink</u>	 Sí	 Sí	 No
<u>AWS::RTBFabric::Link</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::RTBFabric::OutboundExternalLink</u>	 Sí	 Sí	 No
<u>AWS::RTBFabric::RequesterGateway</u>	 Sí	 Sí	 No
<u>AWS::RTBFabric::ResponderGateway</u>	 Sí	 Sí	 No
<u>AWS::RUM::AppMonitor</u>	 Sí	 Sí	 Sí
<u>AWS::Rbin::Rule</u>	 Sí	 Sí	 No
<u>AWS::Redshift::Cluster</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::Redshift::ClusterParameterGroup</u>	 Sí	 Sí	 No
<u>AWS::Redshift::ClusterSubnetGroup</u>	 Sí	 Sí	 No
<u>AWS::Redshift::EndpointAccess</u>	 Sí	 Sí	 Sí
<u>AWS::Redshift::EndpointAuthorization</u>	 Sí	 Sí	 Sí
<u>AWS::Redshift::EventSubscription</u>	 Sí	 Sí	 No
<u>AWS::Redshift::Integration</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::Redshift::ScheduledAction</u>	 Sí	 Sí	 Sí
<u>AWS::RedshiftServerless::Namespace</u>	 Sí	 Sí	 No
<u>AWS::RedshiftServerless::Snapshot</u>	 Sí	 Sí	 No
<u>AWS::RedshiftServerless::Workgroup</u>	 Sí	 Sí	 No
<u>AWS::RefactorSpaces::Application</u>	 Sí	 Sí	 Sí
<u>AWS::RefactorSpaces::Environment</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::RefactorSpaces::Route</u>	 Sí	 Sí	 Sí
<u>AWS::RefactorSpaces::Service</u>	 Sí	 Sí	 Sí
<u>AWS::Rekognition::Collection</u>	 Sí	 Sí	 Sí
<u>AWS::Rekognition::Project</u>	 Sí	 Sí	 Sí
<u>AWS::Rekognition::StreamProcessor</u>	 Sí	 Sí	 Sí
<u>AWS::ResilienceHub::App</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::ResilienceHub::ResiliencyPolicy</u>	 Sí	 Sí	 Sí
<u>AWS::ResourceExplorer2::DefaultViewAssociation</u>	 Sí	 Sí	 No
<u>AWS::ResourceExplorer2::Index</u>	 Sí	 Sí	 Sí
<u>AWS::ResourceExplorer2::View</u>	 Sí	 Sí	 Sí
<u>AWS::ResourceGroups::Group</u>	 Sí	 Sí	 Sí
<u>AWS::ResourceGroups::TagSyncTask</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
AWS::RolesAnywhere::CRL	 Sí	 Sí	 Sí
AWS::RolesAnywhere::Profile	 Sí	 Sí	 Sí
AWS::RolesAnywhere::TrustAnchor	 Sí	 Sí	 No
AWS::Route53::CidrCollection	 Sí	 Sí	 Sí
AWS::Route53::DNSSEC	 Sí	 Sí	 Sí
AWS::Route53::HealthCheck	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::Route53::HostedZone</u>	 Sí	 Sí	 Sí
<u>AWS::Route53::KeySigningKey</u>	 Sí	 Sí	 Sí
<u>AWS::Route53Profiles::Profile</u>	 Sí	 Sí	 No
<u>AWS::Route53Profiles::ProfileAssociation</u>	 Sí	 Sí	 No
<u>AWS::Route53Profiles::ProfileResourceAssociation</u>	 Sí	 Sí	 No
<u>AWS::Route53RecoveryControl::Cluster</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::Route53RecoveryControl::ControlPanel</u>	 Sí	 Sí	 Sí
<u>AWS::Route53RecoveryControl::RoutingControl</u>	 Sí	 Sí	 Sí
<u>AWS::Route53RecoveryControl::SafetyRule</u>	 Sí	 Sí	 Sí
<u>AWS::Route53RecoveryReadiness::Cell</u>	 Sí	 Sí	 Sí
<u>AWS::Route53RecoveryReadiness::ReadinessCheck</u>	 Sí	 Sí	 Sí
<u>AWS::Route53RecoveryReadiness::RecoveryGroup</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::Route53RecoveryReadiness::ResourceSet</u>	 Sí	 Sí	 Sí
<u>AWS::Route53Resolver::FirewallDomainList</u>	 Sí	 Sí	 No
<u>AWS::Route53Resolver::FirewallRuleGroup</u>	 Sí	 Sí	 Sí
<u>AWS::Route53Resolver::FirewallRuleGroupAssociation</u>	 Sí	 Sí	 Sí
<u>AWS::Route53Resolver::OutpostResolver</u>	 Sí	 Sí	 Sí
<u>AWS::Route53Resolver::ResolverConfig</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::Route53Resolver::ResolverDNSSECConfig</u>	 Sí	 Sí	 Sí
<u>AWS::Route53Resolver::ResolverEndpoint</u>	 Sí	 Sí	 No
<u>AWS::Route53Resolver::ResolverQueryLoggingConfig</u>	 Sí	 Sí	 Sí
<u>AWS::Route53Resolver::ResolverQueryLoggingConfigAssociation</u>	 Sí	 Sí	 Sí
<u>AWS::Route53Resolver::ResolverRule</u>	 Sí	 Sí	 No
<u>AWS::Route53Resolver::ResolverRuleAssociation</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::S3::AccessGrant</u>	 Sí	 Sí	 No
<u>AWS::S3::AccessGrantsInstance</u>	 Sí	 Sí	 No
<u>AWS::S3::AccessGrantsLocation</u>	 Sí	 Sí	 No
<u>AWS::S3::AccessPoint</u>	 Sí	 Sí	 Sí
<u>AWS::S3::Bucket</u>	 Sí	 Sí	 Sí
<u>AWS::S3::BucketPolicy</u>	 Sí	 No	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::S3::MultiRegionAccessPoint</u>	 Sí	 Sí	 Sí
<u>AWS::S3::MultiRegionAccessPointPolicy</u>	 Sí	 No	 No
<u>AWS::S3::StorageLens</u>	 Sí	 Sí	 Sí
<u>AWS::S3::StorageLensGroup</u>	 Sí	 Sí	 No
<u>AWS::S3Express::AccessPoint</u>	 Sí	 Sí	 No
<u>AWS::S3Express::BucketPolicy</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::S3Express::DirectoryBucket</u>	 Sí	 Sí	 No
<u>AWS::S3ObjectLambda::AccessPoint</u>	 Sí	 Sí	 Sí
<u>AWS::S3ObjectLambda::AccessPointPolicy</u>	 Sí	 Sí	 No
<u>AWS::S3Outposts::AccessPoint</u>	 Sí	 Sí	 Sí
<u>AWS::S3Outposts::Bucket</u>	 Sí	 Sí	 Sí
<u>AWS::S3Outposts::BucketPolicy</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::S3Outposts::Endpoint</u>	 Sí	 Sí	 Sí
<u>AWS::S3Tables::Namespace</u>	 Sí	 Sí	 No
<u>AWS::S3Tables::Table</u>	 Sí	 Sí	 No
<u>AWS::S3Tables::TableBucket</u>	 Sí	 Sí	 No
<u>AWS::S3Tables::TableBucketPolicy</u>	 Sí	 Sí	 No
<u>AWS::S3Tables::TablePolicy</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::S3Vectors::Index</u>	 Sí	 Sí	 No
<u>AWS::S3Vectors::VectorBucket</u>	 Sí	 Sí	 No
<u>AWS::S3Vectors::VectorBucketPolicy</u>	 Sí	 Sí	 No
<u>AWS::SES::ConfigurationSet</u>	 Sí	 Sí	 No
<u>AWS::SES::ConfigurationSetEventDestination</u>	 No	 Sí	 No
<u>AWS::SES::ContactList</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::SES::DedicatedIpPool</u>	 Sí	 Sí	 Sí
<u>AWS::SES::EmailIdentity</u>	 Sí	 Sí	 Sí
<u>AWS::SES::MailManagerAddonInstance</u>	 Sí	 Sí	 No
<u>AWS::SES::MailManagerAddonSubscription</u>	 Sí	 Sí	 No
<u>AWS::SES::MailManagerAddressList</u>	 Sí	 Sí	 No
<u>AWS::SES::MailManagerArchive</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::SES::MailManagerIngressPoint</u>	 Sí	 Sí	 No
<u>AWS::SES::MailManagerRelay</u>	 Sí	 Sí	 No
<u>AWS::SES::MailManagerRuleSet</u>	 Sí	 Sí	 No
<u>AWS::SES::MailManagerTrafficPolicy</u>	 Sí	 Sí	 No
<u>AWS::SES::MultiRegionEndpoint</u>	 Sí	 Sí	 No
<u>AWS::SES::Template</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::SES::Tenant</u>	 Sí	 Sí	 No
<u>AWS::SES::VdmAttributes</u>	 Sí	 Sí	 No
<u>AWS::SMSVOICE::ConfigurationSet</u>	 Sí	 Sí	 No
<u>AWS::SMSVOICE::OptOutList</u>	 Sí	 Sí	 No
<u>AWS::SMSVOICE::PhoneNumber</u>	 Sí	 Sí	 No
<u>AWS::SMSVOICE::Pool</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::SMSVOICE::ProtectConfiguration</u>	 Sí	 Sí	 No
<u>AWS::SMSVOICE::ResourcePolicy</u>	 Sí	 Sí	 No
<u>AWS::SMSVOICE::SenderId</u>	 Sí	 Sí	 No
<u>AWS::SNS::Subscription</u>	 Sí	 Sí	 No
<u>AWS::SNS::Topic</u>	 Sí	 Sí	 Sí
<u>AWS::SNS::TopicInlinePolicy</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::SQS::Queue</u>	 Sí	 Sí	 Sí
<u>AWS::SQS::QueueInlinePolicy</u>	 Sí	 Sí	 No
<u>AWS::SSM::Association</u>	 Sí	 Sí	 Sí
<u>AWS::SSM::Document</u>	 Sí	 No	 No
<u>AWS::SSM::MaintenanceWindowTarget</u>	 Sí	 Sí	 No
<u>AWS::SSM::MaintenanceWindowTask</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::SSM::Parameter</u>	 Sí	 Sí	 No
<u>AWS::SSM::PatchBaseline</u>	 Sí	 Sí	 No
<u>AWS::SSM::ResourceDataSync</u>	 Sí	 Sí	 No
<u>AWS::SSM::ResourcePolicy</u>	 Sí	 Sí	 Sí
<u>AWS::SSMContacts::Contact</u>	 Sí	 Sí	 Sí
<u>AWS::SSMContacts::ContactChannel</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::SSMContacts::Plan</u>	 Sí	 Sí	 No
<u>AWS::SSMContacts::Rotation</u>	 Sí	 Sí	 No
<u>AWS::SSMGuiConnect::Preferences</u>	 Sí	 Sí	 No
<u>AWS::SSMIncidents::ReplicationSet</u>	 Sí	 Sí	 Sí
<u>AWS::SSMIncidents::ResponsePlan</u>	 Sí	 Sí	 Sí
<u>AWS::SSMQuickSetup::ConfigurationManager</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::SSMQuickSetup::LifecycleAutomation</u>	 Sí	 Sí	 No
<u>AWS::SSO::Application</u>	 Sí	 Sí	 No
<u>AWS::SSO::ApplicationAssignment</u>	 Sí	 Sí	 No
<u>AWS::SSO::Assignment</u>	 Sí	 Sí	 No
<u>AWS::SSO::Instance</u>	 Sí	 Sí	 No
<u>AWS::SSO::InstanceAccessControlAttributeConfiguration</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::SageMaker::App</u>	 Sí	 Sí	 No
<u>AWS::SageMaker::AppImageConfig</u>	 Sí	 Sí	 No
<u>AWS::SageMaker::Cluster</u>	 Sí	 Sí	 No
<u>AWS::SageMaker::DataQualityJobDefinition</u>	 Sí	 Sí	 No
<u>AWS::SageMaker::Device</u>	 Sí	 Sí	 No
<u>AWS::SageMaker::DeviceFleet</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::SageMaker::Domain</u>	 Sí	 Sí	 No
<u>AWS::SageMaker::Endpoint</u>	 No	 Sí	 No
<u>AWS::SageMaker::FeatureGroup</u>	 Sí	 Sí	 Sí
<u>AWS::SageMaker::Image</u>	 Sí	 Sí	 Sí
<u>AWS::SageMaker::ImageVersion</u>	 Sí	 Sí	 Sí
<u>AWS::SageMaker::InferenceComponent</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::SageMaker::InferenceExperiment</u>	 Sí	 Sí	 Sí
<u>AWS::SageMaker::MlflowTrackingServer</u>	 Sí	 Sí	 No
<u>AWS::SageMaker::ModelBiasJobDefinition</u>	 Sí	 Sí	 No
<u>AWS::SageMaker::ModelCard</u>	 Sí	 Sí	 Sí
<u>AWS::SageMaker::ModelExplainabilityJobDefinition</u>	 Sí	 Sí	 No
<u>AWS::SageMaker::ModelPackage</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::SageMaker::ModelPackageGroup</u>	 Sí	 Sí	 Sí
<u>AWS::SageMaker::ModelQualityJobDefinition</u>	 Sí	 Sí	 No
<u>AWS::SageMaker::MonitoringSchedule</u>	 Sí	 Sí	 No
<u>AWS::SageMaker::PartnerApp</u>	 Sí	 Sí	 No
<u>AWS::SageMaker::Pipeline</u>	 Sí	 Sí	 Sí
<u>AWS::SageMaker::ProcessingJob</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::SageMaker::Project</u>	 Sí	 Sí	 Sí
<u>AWS::SageMaker::Space</u>	 Sí	 Sí	 No
<u>AWS::SageMaker::StudioLifecycleConfig</u>	 Sí	 Sí	 No
<u>AWS::SageMaker::UserProfile</u>	 Sí	 Sí	 No
<u>AWS::Scheduler::Schedule</u>	 Sí	 Sí	 Sí
<u>AWS::Scheduler::ScheduleGroup</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::SecretsManager::ResourcePolicy</u>	 Sí	 No	 No
<u>AWS::SecretsManager::RotationSchedule</u>	 Sí	 Sí	 No
<u>AWS::SecretsManager::Secret</u>	 Sí	 Sí	 Sí
<u>AWS::SecretsManager::SecretTargetAttachment</u>	 Sí	 Sí	 No
<u>AWS::SecurityHub::AggregatorV2</u>	 Sí	 Sí	 No
<u>AWS::SecurityHub::AutomationRule</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::SecurityHub::AutomationRuleV2</u>	 Sí	 Sí	 No
<u>AWS::SecurityHub::ConfigurationPolicy</u>	 Sí	 Sí	 No
<u>AWS::SecurityHub::ConnectorV2</u>	 Sí	 Sí	 No
<u>AWS::SecurityHub::DelegatedAdmin</u>	 Sí	 Sí	 No
<u>AWS::SecurityHub::FindingAggregator</u>	 Sí	 Sí	 No
<u>AWS::SecurityHub::Hub</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::SecurityHub::HubV2</u>	 Sí	 Sí	 No
<u>AWS::SecurityHub::Insight</u>	 Sí	 Sí	 No
<u>AWS::SecurityHub::OrganizationConfiguration</u>	 Sí	 Sí	 No
<u>AWS::SecurityHub::PolicyAssociation</u>	 Sí	 Sí	 No
<u>AWS::SecurityHub::ProductSubscription</u>	 Sí	 Sí	 No
<u>AWS::SecurityHub::SecurityControl</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::SecurityHub::Standard</u>	 Sí	 Sí	 No
<u>AWS::SecurityLake::AwsLogSource</u>	 Sí	 Sí	 No
<u>AWS::SecurityLake::DataLake</u>	 Sí	 Sí	 No
<u>AWS::SecurityLake::Subscriber</u>	 Sí	 Sí	 No
<u>AWS::SecurityLake::SubscriberNotification</u>	 Sí	 Sí	 No
<u>AWS::ServiceCatalog::CloudFormationProvisionedProduct</u>	 Sí	 No	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::ServiceCatalog::LaunchNotificationConstraint</u>	 Sí	 Sí	 No
<u>AWS::ServiceCatalog::LaunchTemplateConstraint</u>	 Sí	 Sí	 No
<u>AWS::ServiceCatalog::PortfolioPrincipalAssociation</u>	 Sí	 Sí	 No
<u>AWS::ServiceCatalog::PortfolioProductAssociation</u>	 Sí	 Sí	 No
<u>AWS::ServiceCatalog::PortfolioShare</u>	 Sí	 Sí	 No
<u>AWS::ServiceCatalog::ResourceUpdateConstraint</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::ServiceCatalog::ServiceAction</u>	 Sí	 Sí	 Sí
<u>AWS::ServiceCatalog::ServiceActionAssociation</u>	 Sí	 Sí	 No
<u>AWS::ServiceCatalog::TagOption</u>	 Sí	 Sí	 No
<u>AWS::ServiceCatalog::TagOptionAssociation</u>	 Sí	 Sí	 No
<u>AWS::ServiceCatalogAppRegistry::Application</u>	 Sí	 Sí	 Sí
<u>AWS::ServiceCatalogAppRegistry::AttributeGroup</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::ServiceCatalogAppRegistry::AttributeGroupAssociation</u>	 Sí	 Sí	 No
<u>AWS::ServiceCatalogAppRegistry::ResourceAssociation</u>	 Sí	 No	 No
<u>AWS::Shield::DRTAccess</u>	 Sí	 Sí	 No
<u>AWS::Shield::ProactiveEngagement</u>	 Sí	 Sí	 No
<u>AWS::Shield::Protection</u>	 Sí	 Sí	 No
<u>AWS::Shield::ProtectionGroup</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::Signer::ProfilePermission</u>	 Sí	 Sí	 No
<u>AWS::Signer::SigningProfile</u>	 Sí	 Sí	 Sí
<u>AWS::SimSpaceWeaver::Simulation</u>	 Sí	 Sí	 Sí
<u>AWS::StepFunctions::Activity</u>	 Sí	 Sí	 No
<u>AWS::StepFunctions::StateMachine</u>	 Sí	 Sí	 Sí
<u>AWS::StepFunctions::StateMachineAlias</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::StepFunctions::StateMachineVersion</u>	 Sí	 Sí	 Sí
<u>AWS::SupportApp::AccountAlias</u>	 Sí	 Sí	 Sí
<u>AWS::SupportApp::SlackChannelConfiguration</u>	 Sí	 Sí	 Sí
<u>AWS::SupportApp::SlackWorkspaceConfiguration</u>	 Sí	 Sí	 Sí
<u>AWS::Synthetics::Canary</u>	 Sí	 Sí	 Sí
<u>AWS::Synthetics::Group</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::SystemsManagerSAP::Application</u>	 Sí	 Sí	 No
<u>AWS::Timestream::Database</u>	 Sí	 Sí	 Sí
<u>AWS::Timestream::InfluxDBInstance</u>	 Sí	 Sí	 No
<u>AWS::Timestream::ScheduledQuery</u>	 Sí	 Sí	 No
<u>AWS::Timestream::Table</u>	 Sí	 Sí	 Sí
<u>AWS::Transfer::Agreement</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::Transfer::Certificate</u>	 Sí	 Sí	 Sí
<u>AWS::Transfer::Connector</u>	 Sí	 Sí	 Sí
<u>AWS::Transfer::Profile</u>	 Sí	 Sí	 Sí
<u>AWS::Transfer::Server</u>	 Sí	 Sí	 No
<u>AWS::Transfer::User</u>	 Sí	 Sí	 No
<u>AWS::Transfer::WebApp</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::Transfer::Workflow</u>	 Sí	 Sí	 Sí
<u>AWS::VerifiedPermissions::IdentitySource</u>	 Sí	 Sí	 Sí
<u>AWS::VerifiedPermissions::Policy</u>	 Sí	 Sí	 Sí
<u>AWS::VerifiedPermissions::PolicyStore</u>	 Sí	 Sí	 Sí
<u>AWS::VerifiedPermissions::PolicyTemplate</u>	 Sí	 Sí	 Sí
<u>AWS::VoiceID::Domain</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::VpcLattice::AccessLogSubscription</u>	 Sí	 Sí	 Sí
<u>AWS::VpcLattice::AuthPolicy</u>	 Sí	 Sí	 No
<u>AWS::VpcLattice::DomainVerification</u>	 Sí	 Sí	 No
<u>AWS::VpcLattice::Listener</u>	 Sí	 Sí	 Sí
<u>AWS::VpcLattice::ResourceConfiguration</u>	 Sí	 Sí	 No
<u>AWS::VpcLattice::ResourceGateway</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::VpcLattice::ResourcePolicy</u>	 Sí	 Sí	 No
<u>AWS::VpcLattice::Rule</u>	 Sí	 Sí	 Sí
<u>AWS::VpcLattice::Service</u>	 Sí	 Sí	 Sí
<u>AWS::VpcLattice::ServiceNetwork</u>	 Sí	 Sí	 Sí
<u>AWS::VpcLattice::ServiceNetworkResourceAssociation</u>	 Sí	 Sí	 No
<u>AWS::VpcLattice::ServiceNetworkServiceAssociation</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::VpcLattice::ServiceNetworkVpcAssociation</u>	 Sí	 Sí	 Sí
<u>AWS::VpcLattice::TargetGroup</u>	 Sí	 Sí	 Sí
<u>AWS::WAFv2::IPSet</u>	 Sí	 Sí	 Sí
<u>AWS::WAFv2::LoggingConfiguration</u>	 Sí	 Sí	 Sí
<u>AWS::WAFv2::RegexPatternSet</u>	 Sí	 Sí	 Sí
<u>AWS::WAFv2::RuleGroup</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::WAFv2::WebACL</u>	 Sí	 Sí	 Sí
<u>AWS::WAFv2::WebACLAssociation</u>	 Sí	 Sí	 No
<u>AWS::Wisdom::AIAgent</u>	 Sí	 Sí	 No
<u>AWS::Wisdom::AIAgentVersion</u>	 Sí	 Sí	 No
<u>AWS::Wisdom::AIGuardrail</u>	 Sí	 Sí	 No
<u>AWS::Wisdom::AIGuardrailVersion</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::Wisdom::AIPrompt</u>	 Sí	 Sí	 No
<u>AWS::Wisdom::AIPromptVersion</u>	 Sí	 Sí	 No
<u>AWS::Wisdom::Assistant</u>	 Sí	 Sí	 Sí
<u>AWS::Wisdom::AssistantAssociation</u>	 Sí	 Sí	 Sí
<u>AWS::Wisdom::KnowledgeBase</u>	 Sí	 Sí	 Sí
<u>AWS::Wisdom::MessageTemplate</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::Wisdom::MessageTemplateVersion</u>	 Sí	 Sí	 No
<u>AWS::Wisdom::QuickResponse</u>	 Sí	 Sí	 No
<u>AWS::WorkSpaces::ConnectionAlias</u>	 Sí	 Sí	 No
<u>AWS::WorkSpaces::WorkspacesPool</u>	 Sí	 Sí	 No
<u>AWS::WorkSpacesThinClient::Environment</u>	 Sí	 Sí	 No
<u>AWS::WorkSpacesWeb::BrowserSettings</u>	 Sí	 Sí	 Sí

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::WorkSpacesWeb::DataProtectionSettings</u>	 Sí	 Sí	 No
<u>AWS::WorkSpacesWeb::IdentityProvider</u>	 Sí	 Sí	 Sí
<u>AWS::WorkSpacesWeb::IpAccessSettings</u>	 Sí	 Sí	 Sí
<u>AWS::WorkSpacesWeb::NetworkSettings</u>	 Sí	 Sí	 Sí
<u>AWS::WorkSpacesWeb::Portal</u>	 Sí	 Sí	 Sí
<u>AWS::WorkSpacesWeb::SessionLogger</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::WorkSpacesWeb::TrustStore</u>	 Sí	 Sí	 Sí
<u>AWS::WorkSpacesWeb::UserAccessLoggingSettings</u>	 Sí	 Sí	 Sí
<u>AWS::WorkSpacesWeb::UserSettings</u>	 Sí	 Sí	 Sí
<u>AWS::WorkspacesInstances::Volume</u>	 Sí	 Sí	 No
<u>AWS::WorkspacesInstances::VolumeAssociation</u>	 Sí	 Sí	 No
<u>AWS::WorkspacesInstances::WorkspaceInstance</u>	 Sí	 Sí	 No

Recurso	Importación	Detección de desviaciones	Generador de laC
<u>AWS::XRay::Group</u>	 Sí	 Sí	 Sí
<u>AWS::XRay::ResourcePolicy</u>	 Sí	 Sí	 Sí
<u>AWS::XRay::SamplingRule</u>	 Sí	 Sí	 No
<u>AWS::XRay::TransactionSearchConfig</u>	 Sí	 Sí	 No

Uso de los enlaces de creación rápida para crear pilas de CloudFormation

Los enlaces de creación rápida proporcionan un método simplificado para inicializar pilas de CloudFormation directamente desde URL en la consola de CloudFormation. Al especificar la URL de la plantilla, el nombre de la pila y los parámetros de las plantillas como parámetros de consulta de URL, puede rellenar previamente una única página Crear pila y acelerar la creación de la pila. Esto simplifica el proceso de creación de pilas al reducir tanto el número de páginas del asistente como la cantidad de datos que debe introducir el usuario. También optimiza la reutilización de plantillas, ya que puede crear varias URL que especifiquen diferentes valores para la misma plantilla.

Formato de URL

El enlace de creación rápida sigue este formato de URL:

```
https://region-code.console.aws.amazon.com/cloudformation/home?region=region-code#/  
stacks/create/review  
?templateURL=TemplateURL  
&stackName=StackName  
&param_parameterName=parameterValue
```

CloudFormation es compatible con los siguientes parámetros de consulta de URL:

URL de plantilla

Obligatorio. El parámetro `templateURL` especifica la URL de la plantilla de pila ubicada en un bucket de Amazon S3. Para evitar problemas de acceso con una URL prefirmada de S3, asegúrese de codificar la URL en formato URL.

Formatos de URL de S3 compatibles:

- `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` (formato heredado)

Nombre de pila

Opcional. Utilice el parámetro `stackName` para especificar el nombre de la pila de CloudFormation que se va a crear. El nombre de una pila puede contener únicamente caracteres alfanuméricos (con distinción entre mayúsculas y minúsculas) y guiones. Debe comenzar con un carácter alfabético y no puede tener más de 128 caracteres.

Parámetros de plantilla

Opcional. Para los parámetros de la plantilla de pila que no sean de tipo `NoEcho`, utilice el formato `param_parameterName` en la cadena de consulta de la URL. El parámetro de la URL debe incluir el prefijo `param_` y el segmento del nombre del parámetro debe coincidir exactamente con el nombre del parámetro de la plantilla. Por ejemplo: `param_DBName`.

CloudFormation no tiene en cuenta los parámetros que no existen en la plantilla ni los tipos de parámetros definidos con la propiedad `NoEcho` establecida en `true` (normalmente, nombres

de usuarios y contraseñas). Los parámetros de URL invalidan los valores predeterminados especificados en la plantilla. Puede incluir tantos parámetros como sean necesarios.

Important

En lugar de integrar información confidencial directamente en las plantillas de CloudFormation, se recomienda utilizar parámetros dinámicos en la plantilla de la pila para hacer referencia a la información confidencial almacenada y administrada fuera de CloudFormation, como en AWS Systems Manager Parameter Store o AWS Secrets Manager.

Para obtener más información, consulte la práctica recomendada [No integre credenciales en sus plantillas](#).

Todos los nombres de parámetros de consulta distinguen entre mayúsculas y minúsculas. Los usuarios pueden sobrescribir estos valores en la consola antes de crear la pila.

Ejemplo

El siguiente ejemplo se basa en la plantilla de muestra de [instancia básica única de WordPress](#). La cadena de consulta incluye el parámetro `templateURL` necesario y los parámetros `stackName`, `DBName`, `InstanceType` y `KeyName`.

Se agregaron saltos de línea en la siguiente dirección URL para mayor claridad.

```
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
```

La siguiente dirección URL incluye los mismos parámetros que el ejemplo anterior, pero sin saltos de línea. Este es el formato de URL real.

```
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=
```

Creación de una pila mediante un enlace de creación rápida

Al abrir un enlace de creación rápida, se lo dirige a la consola de CloudFormation. La consola se abre directamente en la página Crear pila rápidamente, con los valores suministrados utilizados automáticamente para los parámetros.

Cómo crear una pila mediante un enlace de creación rápida (consola)

1. En la página Creación rápida de pila, en Plantilla, URL de plantilla, confirme que la URL de la plantilla es correcta.
2. Expanda la sección Ver plantilla para verificar la plantilla.
3. En Nombre de pila, verifique el nombre de pila rellenado previamente.
4. Revise la sección Parámetros. Verifique que los valores de los parámetros rellenado previamente sean correctos. Rellene los parámetros obligatorios que no se hayan especificado en la URL. Modifique los valores rellenados previamente si es necesario.
5. A continuación, puede configurar los siguientes ajustes:
 - Etiquetas: organice los recursos con pares clave-valor.
 - Permisos: elija el rol de servicio de IAM para las operaciones de pila.
 - Opciones de error de pila: elija entre revertir (predeterminado) o conservar los recursos.
 - Política de pila: controle los permisos de actualización de recursos.
 - Configuración de reversión: configure la reversión basada en alarmas de CloudWatch.
 - Opciones de notificación: configure las notificaciones de Amazon SNS para eventos de pila.
 - Opciones de creación de pila: defina el tiempo máximo de creación de pila y habilite la protección contra terminación para evitar eliminaciones accidentales.

Para obtener más información, consulte [Configurar las opciones la pila](#).

6. En Capacidades, complete los reconocimientos requeridos. Por ejemplo, si la plantilla contiene recursos de IAM, seleccione Confirmando que esta plantilla puede crear recursos de IAM para especificar que desea utilizar recursos de IAM en la plantilla. Para obtener más información, consulte [Reconocimiento de recursos de IAM en plantillas de CloudFormation](#).
7. (Opcional) Puede crear un conjunto de cambios para obtener una vista previa de la configuración de la pila antes de crearla. Elija Crear conjunto de cambios y siga las instrucciones. Para obtener más información, consulte [Obtenga una vista previa de la configuración de la pila](#).

8. Cuando esté listo, elija Crear pila para inicializar la pila y luego supervise el progreso de creación de la pila en la pestaña Eventos. Para obtener más información, consulte [Monitorizar el progreso de la pila](#).

Ejemplos de comandos de operación de pila de CloudFormation para la AWS CLI y PowerShell

En los siguientes ejemplos de línea de comandos, se muestra cómo llevar a cabo acciones individuales de CloudFormation con la AWS CLI y PowerShell. Estos ejemplos incluyen solo las acciones que se utilizan con mayor frecuencia. Para ver una lista completa, consulte [cloudformation](#) en la Referencia de los comandos de AWS CLI.

En los ejemplos de esta guía se utiliza la barra invertida (\) como convención para indicar que una línea de comandos larga continúa en la línea siguiente.

Temas

- [Cancelar una actualización de pila](#)
- [Continuación de la restauración de una actualización](#)
- [Creación de una pila](#)
- [Crear una pila que incluya transformaciones](#)
- [Eliminar una pila](#)
- [Describir los eventos de la pila](#)
- [Describir un recurso de pila](#)
- [Describir los recursos de pila](#)
- [Describir las pilas](#)
- [Obtener una plantilla](#)
- [Recursos de pila de listas](#)
- [Listar pilas](#)
- [Actualizar una pila](#)
- [Valide su plantillas](#)
- [Carga de artefactos locales en un bucket de S3 con la AWS CLI](#)

Cancelar una actualización de pila

Utilice el comando [cancel-update-stack](#) para cancelar una actualización de pila. Para obtener más información, consulte [Cancelar una actualización de pila](#).

CLI

AWS CLI

Cancelación de una actualización de pila en progreso

El comando `cancel-update-stack` a continuación cancela una actualización de pila en la pila `myteststack`:

```
aws cloudformation cancel-update-stack --stack-name myteststack
```

- Para ver detalles de la API, consulte [CancelUpdateStack](#) en la Referencia de comandos de la AWS CLI.

PowerShell

Herramientas para PowerShell V4

Ejemplo 1: Cancelación de una actualización en la pila especificada.

```
Stop-CFNUpdateStack -StackName "myStack"
```

- Para obtener información sobre la API, consulte [CancelUpdateStack](#) en Herramientas de AWS para PowerShell Cmdlet Reference (V4).

Herramientas para PowerShell V5

Ejemplo 1: Cancelación de una actualización en la pila especificada.

```
Stop-CFNUpdateStack -StackName "myStack"
```

- Para obtener información sobre la API, consulte [CancelUpdateStack](#) en Herramientas de AWS para PowerShell Cmdlet Reference (V5).

Continuación de la restauración de una actualización

Utilice el comando [continue-update-rollback](#) para continuar la restauración de una actualización. Para obtener más información, consulte [Continuación de la restauración de una actualización](#).

CLI

AWS CLI

Nuevo intento de reversión de actualizaciones

El ejemplo de `continue-update-rollback` a continuación reanuda una operación de reversión a partir de un intento fallido previo de una actualización de pila.

```
aws cloudformation continue-update-rollback \  
  --stack-name my-stack
```

Este comando no genera ninguna salida.

- Para obtener información sobre la API, consulte [ContinueUpdateRollback](#) en la Referencia de comandos de la AWS CLI.

PowerShell

Herramientas para PowerShell V4

Ejemplo 1: Reanudación de la reversión de la pila seleccionada, la cual debería encontrarse en el estado "UPDATE_ROLLBACK_FAILED". Si la reversión continua es exitosa, el estado de la pila será "UPDATE_ROLLBACK_COMPLETE".

```
Resume-CFNUdateRollback -StackName "myStack"
```

- Para obtener información sobre la API, consulte [ContinueUpdateRollback](#) en Herramientas de AWS para PowerShell Cmdlet Reference (V4).

Herramientas para PowerShell V5

Ejemplo 1: Reanudación de la reversión de la pila seleccionada, la cual debería encontrarse en el estado "UPDATE_ROLLBACK_FAILED". Si la reversión continua es exitosa, el estado de la pila será "UPDATE_ROLLBACK_COMPLETE".

```
Resume-CFNUpdateRollback -StackName "myStack"
```

- Para obtener información sobre la API, consulte [ContinueUpdateRollback](#) en Herramientas de AWS para PowerShell Cmdlet Reference (V5).

Creación de una pila

Utilice el comando [create-stack](#) para crear una pila. Debe proporcionar el nombre de la pila, la ubicación de una plantilla válida y cualquier parámetro de entrada. Los nombres clave de los parámetros distinguen entre mayúsculas y minúsculas. Si escribe incorrectamente el nombre de la clave de un parámetro, CloudFormation no crea la pila e informa que la plantilla no contiene dicho parámetro.

Los siguientes ejemplos muestran cómo crear una pila nueva con el nombre, la plantilla y los parámetros de entrada especificados.

CLI

AWS CLI

Creación de una pila de AWS CloudFormation

En el siguiente ejemplo de `create-stacks`, se crea una pila con el nombre `myteststack` mediante la plantilla `sampletemplate.json`:

```
aws cloudformation create-stack --stack-name myteststack --template-body file://sampletemplate.json --parameters ParameterKey=KeyPairName,ParameterValue=TestKey  
ParameterKey=SubnetIDs,ParameterValue=SubnetID1\\,SubnetID2
```

Salida:

```
{  
  "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/  
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896"  
}
```

Para obtener más información, consulte Pilas en la Guía del usuario de AWS CloudFormation.

- Para obtener información sobre la API, consulte [CreateStack](#) en la Referencia de comandos de la AWS CLI.

PowerShell

Herramientas para PowerShell V4

Ejemplo 1: Creación de una pila nueva con el nombre especificado. La plantilla se analiza a partir del contenido suministrado con los parámetros de personalización ("PK1" y "PK2" representan los nombres de los parámetros declarados en el contenido de la plantilla, mientras que "PV1" y "PV2" representan los valores de dichos parámetros). Los parámetros de personalización también se pueden especificar con "Key" y "Value" en lugar de "ParameterKey" y "ParameterValue". Si la creación de la pila falla, no se revertirá.

```
New-CFNStack -StackName "myStack" `
              -TemplateBody "{TEMPLATE CONTENT HERE}" `
              -Parameter @( @{ ParameterKey="PK1"; ParameterValue="PV1" },
                             @{ ParameterKey="PK2"; ParameterValue="PV2" }) `
              -DisableRollback $true
```

Ejemplo 2: Creación de una pila nueva con el nombre especificado. La plantilla se analiza a partir del contenido suministrado con los parámetros de personalización ("PK1" y "PK2" representan los nombres de los parámetros declarados en el contenido de la plantilla, mientras que "PV1" y "PV2" representan los valores de dichos parámetros). Los parámetros de personalización también se pueden especificar con "Key" y "Value" en lugar de "ParameterKey" y "ParameterValue". Si la creación de la pila falla, se revertirá.

```
$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"
```

Ejemplo 3: Creación de una pila nueva con el nombre especificado. La plantilla obtenida de la URL de Amazon S3 con los parámetros de personalización ("PK1" representa el nombre del parámetro declarado en el contenido de la plantilla, mientras que "PV1" representa el valor

del parámetro). Los parámetros de personalización también se pueden especificar con “Key” y “Value” en lugar de “ParameterKey” y “ParameterValue”. Si la creación de una pila falla, se revertirá (al igual que al especificar “-DisableRollback \$false”).

```
New-CFNStack -StackName "myStack" `
              -TemplateURL https://s3.amazonaws.com/amzn-s3-demo-bucket/
templatefile.template `
              -Parameter @{ ParameterKey="PK1"; ParameterValue="PV1" }
```

Ejemplo 4: Creación de una pila nueva con el nombre especificado. La plantilla obtenida de la URL de Amazon S3 con los parámetros de personalización (“PK1” representa el nombre del parámetro declarado en el contenido de la plantilla, mientras que “PV1” representa el valor del parámetro). Los parámetros de personalización también se pueden especificar con “Key” y “Value” en lugar de “ParameterKey” y “ParameterValue”. Si la creación de una pila falla, se revertirá (al igual que al especificar “-DisableRollback \$false”). Los AEN de notificación especificados recibirán los eventos publicados relacionados con la pila.

```
New-CFNStack -StackName "myStack" `
              -TemplateURL https://s3.amazonaws.com/amzn-s3-demo-bucket/
templatefile.template `
              -Parameter @{ ParameterKey="PK1"; ParameterValue="PV1" } `
              -NotificationARN @( "arn1", "arn2" )
```

- Para obtener información sobre la API, consulte [CreateStack](#) en Herramientas de AWS para PowerShell Cmdlet Reference (V4).

Herramientas para PowerShell V5

Ejemplo 1: Creación de una pila nueva con el nombre especificado. La plantilla se analiza a partir del contenido suministrado con los parámetros de personalización (“PK1” y “PK2” representan los nombres de los parámetros declarados en el contenido de la plantilla, mientras que “PV1” y “PV2” representan los valores de dichos parámetros). Los parámetros de personalización también se pueden especificar con “Key” y “Value” en lugar de “ParameterKey” y “ParameterValue”. Si la creación de la pila falla, no se revertirá.

```
New-CFNStack -StackName "myStack" `
              -TemplateBody "{TEMPLATE CONTENT HERE}" `
              -Parameter @( @{ ParameterKey="PK1"; ParameterValue="PV1" },
@{ ParameterKey="PK2"; ParameterValue="PV2" }) `
              -DisableRollback $true
```

Ejemplo 2: Creación de una pila nueva con el nombre especificado. La plantilla se analiza a partir del contenido suministrado con los parámetros de personalización (“PK1” y “PK2” representan los nombres de los parámetros declarados en el contenido de la plantilla, mientras que “PV1” y “PV2” representan los valores de dichos parámetros). Los parámetros de personalización también se pueden especificar con “Key” y “Value” en lugar de “ParameterKey” y “ParameterValue”. Si la creación de la pila falla, se revertirá.

```
$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"
```

Ejemplo 3: Creación de una pila nueva con el nombre especificado. La plantilla obtenida de la URL de Amazon S3 con los parámetros de personalización (“PK1” representa el nombre del parámetro declarado en el contenido de la plantilla, mientras que “PV1” representa el valor del parámetro). Los parámetros de personalización también se pueden especificar con “Key” y “Value” en lugar de “ParameterKey” y “ParameterValue”. Si la creación de una pila falla, se revertirá (al igual que al especificar “-DisableRollback \$false”).

```
New-CFNStack -StackName "myStack" `
              -TemplateURL https://s3.amazonaws.com/amzn-s3-demo-bucket/
templatefile.template `
              -Parameter @{ ParameterKey="PK1"; ParameterValue="PV1" }
```

Ejemplo 4: Creación de una pila nueva con el nombre especificado. La plantilla obtenida de la URL de Amazon S3 con los parámetros de personalización (“PK1” representa el nombre del parámetro declarado en el contenido de la plantilla, mientras que “PV1” representa el valor del parámetro). Los parámetros de personalización también se pueden especificar con “Key” y “Value” en lugar de “ParameterKey” y “ParameterValue”. Si la creación de una pila falla, se revertirá (al igual que al especificar “-DisableRollback \$false”). Los AEN de notificación especificados recibirán los eventos publicados relacionados con la pila.

```
New-CFNStack -StackName "myStack" `
              -TemplateURL https://s3.amazonaws.com/amzn-s3-demo-bucket/
templatefile.template `
              -Parameter @{ ParameterKey="PK1"; ParameterValue="PV1" } `
              -NotificationARN @( "arn1", "arn2" )
```

- Para obtener información sobre la API, consulte [CreateStack](#) en Herramientas de AWS para PowerShell Cmdlet Reference (V5).

Note

Puede utilizar la opción `--template-url` de la AWS CLI para especificar la ubicación de un archivo de plantilla en Amazon S3 o AWS Systems Manager.

En el caso de Amazon S3, la URL debe empezar por `https://`. No se admiten las URL de sitios web estáticos de S3.

```
--template-url https://s3.region-code.amazonaws.com/bucket-name/template-name
```

Para AWS Systems Manager, utilice el siguiente formato:

```
--template-url "ssm-doc://arn:aws:ssm:region-code:account-id:document/document-name"
```

Crear una pila que incluya transformaciones

Utilice el comando [deploy](#) para crear una pila que incluya transformaciones. Al crear una pila a partir de una plantilla que incluye transformaciones, debe utilizar un conjunto de cambios. El comando `deploy` combina dos pasos (crear un conjunto de cambios y ejecutarlo) en un solo comando.

AWS CLI

El siguiente comando `deploy` crea una pila denominada con la plantilla especificada.

```
aws cloudformation deploy --stack-name myteststack \
  --template /path_to_template/my-template.json \
  --parameter-overrides Key1=Value1 Key2=Value2
```

Eliminar una pila

Utilice el comando [delete-stack](#) para eliminar una pila. Para obtener más información, consulte [Eliminar una pila](#).

CLI

AWS CLI

Eliminación de una pila

En el siguiente ejemplo de `delete-stack`, se elimina la pila especificada.

```
aws cloudformation delete-stack \  
  --stack-name my-stack
```

Este comando no genera ninguna salida.

- Para obtener información sobre la API, consulte [DeleteStack](#) en la Referencia de comandos de la AWS CLI.

PowerShell

Herramientas para PowerShell V4

Ejemplo 1: Eliminación de la pila especificada.

```
Remove-CFNStack -StackName "myStack"
```

- Para obtener información sobre la API, consulte [DeleteStack](#) en Herramientas de AWS para PowerShell Cmdlet Reference (V4).

Herramientas para PowerShell V5

Ejemplo 1: Eliminación de la pila especificada.

```
Remove-CFNStack -StackName "myStack"
```

- Para obtener información sobre la API, consulte [DeleteStack](#) en Herramientas de AWS para PowerShell Cmdlet Reference (V5).

Si la eliminación falla y devuelve un estado `DELETE_FAILED`, puede optar por eliminar la pila por la fuerza mediante la opción `--deletion-mode`. Estos son los siguientes valores que se pueden utilizar con `deletion-mode`:

- `STANDARD`: elimina la pila de forma normal. Este es el modo de eliminación predeterminado.
- `FORCE_DELETE_STACK`: elimina la pila y omite todos los recursos que no pueden eliminarse.

AWS CLI

El siguiente comando `delete-stack` fuerza la eliminación de la pila *myteststack* mediante el valor `FORCE_DELETE_STACK` con el parámetro `deletion-mode`:

```
aws cloudformation delete-stack --stack-name myteststack \  
  --deletion-mode FORCE_DELETE_STACK
```

Este comando no genera ninguna salida.

Después de usar `FORCE_DELETE_STACK`, puede usar el comando `list-stack-resources` para enumerar los recursos que se omitieron durante el proceso de eliminación de la pila. Los recursos retenidos mostrarán el estado `DELETE_SKIPPED`. Para obtener más información, consulte [Recursos de pila de listas](#).

Describir los eventos de la pila

Utilice el comando [describe-stack-events](#) para describir los eventos de la pila. Para obtener más información, consulte [Monitorizar el progreso de la pila](#).

CLI

AWS CLI

Descripción de los eventos de la pila

En el siguiente ejemplo de `describe-stack-events`, se muestran los dos eventos más recientes de la pila especificada.

```
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=="
}
```

- Para obtener información sobre la API, consulte [DescribeStackEvents](#) en la Referencia de comandos de la AWS CLI.

PowerShell

Herramientas para PowerShell V4

Ejemplo 1: Devolución de todos los eventos relacionados con la pila especificada.

```
Get-CFNStackEvent -StackName "myStack"
```

- Para obtener información sobre la API, consulte [DescribeStackEvents](#) en Herramientas de AWS para PowerShell Cmdlet Reference (V4).

Herramientas para PowerShell V5

Ejemplo 1: Devolución de todos los eventos relacionados con la pila especificada.

```
Get-CFNStackEvent -StackName "myStack"
```

- Para obtener información sobre la API, consulte [DescribeStackEvents](#) en Herramientas de AWS para PowerShell Cmdlet Reference (V5).

Describir un recurso de pila

CLI

AWS CLI

Obtención de información sobre un recurso de pila

En el siguiente ejemplo de `describe-stack-resource`, se muestran los detalles del recurso con el nombre `MyFunction` en la pila especificada.

```
aws cloudformation describe-stack-resource \  
  --stack-name MyStack \  
  --logical-resource-id MyFunction
```

Salida:

```
{  
  "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"
    }
}
```

- Para obtener información sobre la API, consulte [DescribeStackResource](#) en la Referencia de comandos de la AWS CLI.

PowerShell

Herramientas para PowerShell V4

Ejemplo 1: Devolución de la descripción de un recurso identificado en la plantilla asociada a la pila especificada con la identificación lógica "MyDBInstance".

```
Get-CFNStackResource -StackName "myStack" -LogicalResourceId "MyDBInstance"
```

- Para obtener información sobre la API, consulte [DescribeStackResource](#) en Herramientas de AWS para PowerShell Cmdlet Reference (V4).

Herramientas para PowerShell V5

Ejemplo 1: Devolución de la descripción de un recurso identificado en la plantilla asociada a la pila especificada con la identificación lógica "MyDBInstance".

```
Get-CFNStackResource -StackName "myStack" -LogicalResourceId "MyDBInstance"
```

- Para obtener información sobre la API, consulte [DescribeStackResource](#) en Herramientas de AWS para PowerShell Cmdlet Reference (V5).

Describir los recursos de pila

CLI

AWS CLI

Obtención de información sobre un recurso de pila

En el siguiente ejemplo de `describe-stack-resources`, se muestran los detalles de los recursos en la pila especificada.


```
aws cloudformation describe-stack-resources \
  --stack-name my-stack
```

Salida:

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

- Para obtener información sobre la API, consulte [DescribeStackResources](#) en la Referencia de comandos de la AWS CLI.

PowerShell

Herramientas para PowerShell V4

Ejemplo 1: Devolución de las descripciones de los recursos de AWS de hasta 100 recursos asociados a la pila especificada. Para obtener detalles de todos los recursos asociados a una pila, utilice `Get-CFNStackResourceSummary`, que también es compatible con la paginación manual de los resultados.

```
Get-CFNStackResourceList -StackName "myStack"
```

Ejemplo 2: Devolución de la descripción de la instancia de Amazon EC2 identificada en la plantilla asociada a la pila especificada con la identificación lógica "Ec2Instance".

```
Get-CFNStackResourceList -StackName "myStack" -LogicalResourceId "Ec2Instance"
```

Ejemplo 3: Devolución de la descripción de hasta 100 recursos asociados a la pila que contiene una instancia de Amazon EC2 identificada con la identificación de instancia "i-123456". Para obtener detalles de todos los recursos asociados a una pila, utilice `Get-CFNStackResourceSummary`, que también es compatible con la paginación manual de los resultados.

```
Get-CFNStackResourceList -PhysicalResourceId "i-123456"
```

Ejemplo 4: Devolución de la descripción de la instancia de Amazon EC2 identificada con la identificación lógica "Ec2Instance" en la plantilla de una pila. La pila se identifica con la identificación de un recurso físico que también contiene una instancia de Amazon EC2 con la identificación de instancia "i-123456". También se puede utilizar un recurso físico distinto para identificar la pila según el contenido de la plantilla; por ejemplo, un bucket de Amazon S3.

```
Get-CFNStackResourceList -PhysicalResourceId "i-123456" -LogicalResourceId  
"Ec2Instance"
```

- Para obtener información sobre la API, consulte [DescribeStackResources](#) en Herramientas de AWS para PowerShell Cmdlet Reference (V4).

Herramientas para PowerShell V5

Ejemplo 1: Devolución de las descripciones de los recursos de AWS de hasta 100 recursos asociados a la pila especificada. Para obtener detalles de todos los recursos asociados a una pila, utilice Get-CFNStackResourceSummary, que también es compatible con la paginación manual de los resultados.

```
Get-CFNStackResourceList -StackName "myStack"
```

Ejemplo 2: Devolución de la descripción de la instancia de Amazon EC2 identificada en la plantilla asociada a la pila especificada con la identificación lógica "Ec2Instance".

```
Get-CFNStackResourceList -StackName "myStack" -LogicalResourceId "Ec2Instance"
```

Ejemplo 3: Devolución de la descripción de hasta 100 recursos asociados a la pila que contiene una instancia de Amazon EC2 identificada con la identificación de instancia "i-123456". Para obtener detalles de todos los recursos asociados a una pila, utilice Get-CFNStackResourceSummary, que también es compatible con la paginación manual de los resultados.

```
Get-CFNStackResourceList -PhysicalResourceId "i-123456"
```

Ejemplo 4: Devolución de la descripción de la instancia de Amazon EC2 identificada con la identificación lógica "Ec2Instance" en la plantilla de una pila. La pila se identifica con la identificación de un recurso físico que también contiene una instancia de Amazon EC2 con la identificación de instancia "i-123456". También se puede utilizar un recurso físico distinto para identificar la pila según el contenido de la plantilla; por ejemplo, un bucket de Amazon S3.

```
Get-CFNStackResourceList -PhysicalResourceId "i-123456" -LogicalResourceId  
"Ec2Instance"
```

- Para obtener información sobre la API, consulte [DescribeStackResources](#) en Herramientas de AWS para PowerShell Cmdlet Reference (V5).

Describir las pilas

CLI

AWS CLI

Descripción de las pilas de AWS CloudFormation

En el siguiente ejemplo de `describe-stacks`, se muestra la información del resumen para la pila `myteststack`:

```
aws cloudformation describe-stacks --stack-name myteststack
```

Salida:

```
{
  "Stacks": [
    {
      "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896",
      "Description": "AWS CloudFormation Sample Template S3_Bucket: Sample template showing how to create a publicly accessible S3 bucket. **WARNING** This template creates an S3 bucket. You will be billed for the AWS resources used if you create a stack from this template.",
      "Tags": [],
      "Outputs": [
        {
          "Description": "Name of S3 bucket to hold website content",
          "OutputKey": "BucketName",
          "OutputValue": "myteststack-s3bucket-jssofilzie2w"
        }
      ],
      "StackStatusReason": null,
      "CreationTime": "2013-08-23T01:02:15.422Z",
      "Capabilities": [],
      "StackName": "myteststack",
      "StackStatus": "CREATE_COMPLETE",
      "DisableRollback": false
    }
  ]
}
```

Para obtener más información, consulte Pilas en la Guía del usuario de AWS CloudFormation.

- Para obtener información sobre la API, consulte [DescribeStacks](#) en la Referencia de comandos de la AWS CLI.

Go

SDK para Go V2

 Note

Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de 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  
}
```

- Para obtener información sobre la API, consulte [DescribeStacks](#) en la Referencia de la API de AWS SDK para Go.

PowerShell

Herramientas para PowerShell V4

Ejemplo 1: Devolución de una colección de instancias de pila que describen todas las pilas del usuario.

```
Get-CFNStack
```

Ejemplo 2: Devolución de una instancia de pila que describe la pila especificada

```
Get-CFNStack -StackName "myStack"
```

- Para obtener información sobre la API, consulte [DescribeStacks](#) en Herramientas de AWS para PowerShell Cmdlet Reference (V4).

Herramientas para PowerShell V5

Ejemplo 1: Devolución de una colección de instancias de pila que describen todas las pilas del usuario.

```
Get-CFNStack
```

Ejemplo 2: Devolución de una instancia de pila que describe la pila especificada

```
Get-CFNStack -StackName "myStack"
```

- Para obtener información sobre la API, consulte [DescribeStacks](#) en Herramientas de AWS para PowerShell Cmdlet Reference (V5).

De forma predeterminada, el comando `describe-stacks` devuelve los valores de los parámetros. Para evitar la devolución de valores de parámetros confidenciales como contraseñas, incluya una propiedad `NoEcho` establecida en `TRUE` en su plantilla de CloudFormation.

Important

El uso del atributo `NoEcho` no enmascara ninguna información almacenada en lo que se muestra a continuación:

- La sección de la plantilla `Metadata`. CloudFormation no transforma, modifica ni redacta ninguna información que incluya en la sección `Metadata`. Para obtener más información, consulte [Metadata](#).
- La sección de la plantilla `Outputs`. Para obtener más información, consulte [Salidas](#).
- El atributo `Metadata` de una definición de recurso. Para obtener más información, consulte [Metadata atributo](#).

Recomendamos encarecidamente que no utilice estos mecanismos para incluir información confidencial, como contraseñas o secretos.

Important

En lugar de integrar información confidencial directamente en las plantillas de CloudFormation, se recomienda utilizar parámetros dinámicos en la plantilla de la pila para hacer referencia a la información confidencial almacenada y administrada fuera de CloudFormation, como en AWS Systems Manager Parameter Store o AWS Secrets Manager.

Para obtener más información, consulte la práctica recomendada [No integre credenciales en sus plantillas](#).

Obtener una plantilla

CLI

AWS CLI

Visualización del cuerpo de la plantilla de una pila de AWS CloudFormation

En el siguiente ejemplo de `get-template`, se muestra la plantilla para la pila `myteststack`:

```
aws cloudformation get-template --stack-name myteststack
```

Salida:

```
{
  "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"
        }
      }
    }
  }
}
```

- Para obtener información sobre la API, consulte [GetTemplate](#) en la Referencia de comandos de la AWS CLI.

PowerShell

Herramientas para PowerShell V4

Ejemplo 1: Devolución de la plantilla asociada a la pila especificada.

```
Get-CFNTemplate -StackName "myStack"
```

- Para obtener información sobre la API, consulte [GetTemplate](#) en Herramientas de AWS para PowerShell Cmdlet Reference (V4).

Herramientas para PowerShell V5

Ejemplo 1: Devolución de la plantilla asociada a la pila especificada.

```
Get-CFNTemplate -StackName "myStack"
```

- Para obtener información sobre la API, consulte [GetTemplate](#) en Herramientas de AWS para PowerShell Cmdlet Reference (V5).

Recursos de pila de listas

CLI

AWS CLI

Enumeración de los recursos de una pila

En el siguiente comando, se muestra la lista de los recursos en la pila especificada.

```
aws cloudformation list-stack-resources \  
  --stack-name my-stack
```

Salida:

```
{  
  "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"
        }
    }
]
}

```

- Para obtener información sobre la API, consulte [ListStackResources](#) en la Referencia de comandos de la AWS CLI.

PowerShell

Herramientas para PowerShell V4

Ejemplo 1: Devolución de las descripciones de todos los recursos asociados a la pila especificada.

```
Get-CFNStackResourceSummary -StackName "myStack"
```

- Para obtener información sobre la API, consulte [ListStackResources](#) en Herramientas de AWS para PowerShell Cmdlet Reference (V4).

Herramientas para PowerShell V5

Ejemplo 1: Devolución de las descripciones de todos los recursos asociados a la pila especificada.

```
Get-CFNStackResourceSummary -StackName "myStack"
```

- Para obtener información sobre la API, consulte [ListStackResources](#) en Herramientas de AWS para PowerShell Cmdlet Reference (V5).

Listar pilas

Utilice el comando [list-stacks](#) para enumerar las pilas. Para ver únicamente pilas con los códigos de estado especificados, incluya la opción `--stack-status-filter`. Puede especificar uno o varios códigos de estado de pila para la opción `--stack-status-filter`. Para obtener más información, consulte [Códigos de estado de pilas](#).

CLI

AWS CLI

Enumeración de las pilas de AWS CloudFormation

En el siguiente ejemplo de `list-stacks`, se muestra un resumen de todas las pilas que tienen el estado `CREATE_COMPLETE`:

```
aws cloudformation list-stacks --stack-status-filter CREATE_COMPLETE
```

Salida:

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

- Para obtener información sobre la API, consulte [ListStacks](#) en la Referencia de comandos de la AWS CLI.

PowerShell

Herramientas para PowerShell V4

Ejemplo 1: Devolución de información de resumen de todas las pilas.

```
Get-CFNStackSummary
```

Ejemplo 2: Devolución de información de resumen de todas las pilas que se crean actualmente.

```
Get-CFNStackSummary -StackStatusFilter "CREATE_IN_PROGRESS"
```

Ejemplo 3: Devolución de información de resumen de todas las pilas que se crean o se actualizan actualmente.

```
Get-CFNStackSummary -StackStatusFilter @("CREATE_IN_PROGRESS",  
    "UPDATE_IN_PROGRESS")
```

- Para obtener información sobre la API, consulte [ListStacks](#) en Herramientas de AWS para PowerShell Cmdlet Reference (V4).

Herramientas para PowerShell V5

Ejemplo 1: Devolución de información de resumen de todas las pilas.

```
Get-CFNStackSummary
```

Ejemplo 2: Devolución de información de resumen de todas las pilas que se crean actualmente.

```
Get-CFNStackSummary -StackStatusFilter "CREATE_IN_PROGRESS"
```

Ejemplo 3: Devolución de información de resumen de todas las pilas que se crean o se actualizan actualmente.

```
Get-CFNStackSummary -StackStatusFilter @("CREATE_IN_PROGRESS",  
"UPDATE_IN_PROGRESS")
```

- Para obtener información sobre la API, consulte [ListStacks](#) en Herramientas de AWS para PowerShell Cmdlet Reference (V5).

Actualizar una pila

Utilice el comando [update-stack](#) para actualizar directamente una pila. Debe especificar la pila, los valores de los parámetros y las capacidades que desea actualizar y, si desea utilizar una plantilla actualizada, el nombre de la plantilla. Para obtener más información, consulte [Actualizar pilas directamente](#).

CLI

AWS CLI

Actualización de las pilas de AWS CloudFormation

En el siguiente ejemplo de `update-stack`, se actualiza la plantilla y los parámetros de entrada de la pila `mystack`:

```
aws cloudformation update-stack --stack-name mystack --  
template-url https://s3.amazonaws.com/sample/updated.template --  
parameters ParameterKey=KeyPairName,ParameterValue=SampleKeyPair  
ParameterKey=SubnetIDs,ParameterValue=SampleSubnetID1\\,SampleSubnetID2
```

En el siguiente ejemplo de `update-stack`, se actualiza únicamente el valor de parámetro `SubnetIDs` para la pila `mystack`. Si no se especifica un valor de parámetro, se utilizará el valor predeterminado que esté especificado en la plantilla:

```
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
```

En el siguiente ejemplo de `update-stack`, se agregan dos temas de notificaciones de pila en la pila `mystack`:

```
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"
```

Para obtener más información, consulte [Actualizaciones de pila de AWS CloudFormation](#) en la Guía del usuario de AWS CloudFormation.

- Para obtener información sobre la API, consulte [UpdateStack](#) en la Referencia de comandos de la AWS CLI.

PowerShell

Herramientas para PowerShell V4

Ejemplo 1: Actualización de la pila “myStack” con la plantilla y los parámetros de personalización especificados. “PK1” representa el nombre de un parámetro declarado en la plantilla, mientras que “PV1” representa su valor. Los parámetros de personalización también se pueden especificar con “Key” y “Value” en lugar de “ParameterKey” y “ParameterValue”.

```
Update-CFNStack -StackName "myStack" `
    -TemplateBody "{Template Content Here}" `
    -Parameter @{ ParameterKey="PK1"; ParameterValue="PV1" }
```

Ejemplo 2: Actualización de la pila “myStack” con la plantilla y los parámetros de personalización especificados. “PK1” y “PK2” representan los nombres de los parámetros declarados en la plantilla, mientras que “PV1” y “PV2” representan los valores solicitados. Los parámetros de personalización también se pueden especificar con “Key” y “Value” en lugar de “ParameterKey” y “ParameterValue”.

```
Update-CFNStack -StackName "myStack" `
    -TemplateBody "{Template Content Here}" `
    -Parameter @( @{ ParameterKey="PK1"; ParameterValue="PV1" },
    @{ ParameterKey="PK2"; ParameterValue="PV2" } )
```

Ejemplo 3: Actualización de la pila “myStack” con la plantilla y los parámetros de personalización especificados. “PK1” representa el nombre de un parámetro declarado en la

plantilla, mientras que “PV2” representa su valor. Los parámetros de personalización también se pueden especificar con “Key” y “Value” en lugar de “ParameterKey” y “ParameterValue”.

```
Update-CFNStack -StackName "myStack" -TemplateBody "{Template Content Here}" -
Parameters @{ ParameterKey="PK1"; ParameterValue="PV1" }
```

Ejemplo 4: Actualización de la pila “myStack” con la plantilla obtenida de Amazon S3 y los parámetros de personalización especificados. “PK1” y “PK2” representan los nombres de los parámetros declarados en la plantilla, mientras que “PV1” y “PV2” representan los valores solicitados. Los parámetros de personalización también se pueden especificar con “Key” y “Value” en lugar de “ParameterKey” y “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" } )
```

Ejemplo 5: Actualización de la pila “myStack”, que en este ejemplo se supone que contiene recursos de IAM, con la plantilla obtenida de Amazon S3 y los parámetros de personalización específicos. “PK1” y “PK2” representan los nombres de los parámetros declarados en la plantilla, mientras que “PV1” y “PV2” representan los valores solicitados. Los parámetros de personalización también se pueden especificar con “Key” y “Value” en lugar de “ParameterKey” y “ParameterValue”. Las pilas que contienen recursos de IAM requieren que se especifique el parámetro “CAPABILITY_IAM” de -Capabilities; de lo contrario, la actualización fallará y mostrará el error “InsufentCapabilities”.

```
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"
```

- Para obtener información sobre la API, consulte [UpdateStack](#) en Herramientas de AWS para PowerShell Cmdlet Reference (V4).

Herramientas para PowerShell V5

Ejemplo 1: Actualización de la pila “myStack” con la plantilla y los parámetros de personalización especificados. “PK1” representa el nombre de un parámetro declarado en la plantilla, mientras que “PV1” representa su valor. Los parámetros de personalización también se pueden especificar con “Key” y “Value” en lugar de “ParameterKey” y “ParameterValue”.

```
Update-CFNStack -StackName "myStack" `
                -TemplateBody "{Template Content Here}" `
                -Parameter @{ ParameterKey="PK1"; ParameterValue="PV1" }
```

Ejemplo 2: Actualización de la pila “myStack” con la plantilla y los parámetros de personalización especificados. “PK1” y “PK2” representan los nombres de los parámetros declarados en la plantilla, mientras que “PV1” y “PV2” representan los valores solicitados. Los parámetros de personalización también se pueden especificar con “Key” y “Value” en lugar de “ParameterKey” y “ParameterValue”.

```
Update-CFNStack -StackName "myStack" `
                -TemplateBody "{Template Content Here}" `
                -Parameter @( @{ ParameterKey="PK1"; ParameterValue="PV1" },
                             @{ ParameterKey="PK2"; ParameterValue="PV2" } )
```

Ejemplo 3: Actualización de la pila “myStack” con la plantilla y los parámetros de personalización especificados. “PK1” representa el nombre de un parámetro declarado en la plantilla, mientras que “PV2” representa su valor. Los parámetros de personalización también se pueden especificar con “Key” y “Value” en lugar de “ParameterKey” y “ParameterValue”.

```
Update-CFNStack -StackName "myStack" -TemplateBody "{Template Content Here}" -
Parameters @{ ParameterKey="PK1"; ParameterValue="PV1" }
```

Ejemplo 4: Actualización de la pila “myStack” con la plantilla obtenida de Amazon S3 y los parámetros de personalización especificados. “PK1” y “PK2” representan los nombres de los parámetros declarados en la plantilla, mientras que “PV1” y “PV2” representan los valores solicitados. Los parámetros de personalización también se pueden especificar con “Key” y “Value” en lugar de “ParameterKey” y “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" } )
```

Ejemplo 5: Actualización de la pila “myStack”, que en este ejemplo se supone que contiene recursos de IAM, con la plantilla obtenida de Amazon S3 y los parámetros de personalización específicos. “PK1” y “PK2” representan los nombres de los parámetros declarados en la plantilla, mientras que “PV1” y “PV2” representan los valores solicitados. Los parámetros de personalización también se pueden especificar con “Key” y “Value” en lugar de “ParameterKey” y “ParameterValue”. Las pilas que contienen recursos de IAM requieren que se especifique el parámetro “CAPABILITY_IAM” de -Capabilities; de lo contrario, la actualización fallará y mostrará el error “InsufentCapabilities”.

```
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"
```

- Para obtener información sobre la API, consulte [UpdateStack](#) en Herramientas de AWS para PowerShell Cmdlet Reference (V5).

Note

Para eliminar todas las notificaciones, especifique [] en la opción --notification-arns.

Valide su plantilla

Utilice el comando [validate-template](#) para comprobar si hay errores de sintaxis en el archivo de plantilla.

Durante la validación, CloudFormation comprueba primero si la plantilla es un JSON válido. Si no lo es, CloudFormation comprueba si la plantilla es un YAML válido. Si ambas comprobaciones fallan, CloudFormation devuelve un error de validación de plantilla.

CLI

AWS CLI

Validación de una plantilla de AWS CloudFormation

En el siguiente ejemplo de `validate-template`, se valida la plantilla `sampletemplate.json`:

```
aws cloudformation validate-template --template-body file://sampletemplate.json
```

Salida:

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

Para obtener más información, consulte Trabajo con plantillas de AWS CloudFormation en la Guía del usuario de AWS CloudFormation.

- Para obtener información sobre la API, consulte [ValidateTemplate](#) en la Referencia de comandos de la AWS CLI.

PowerShell

Herramientas para PowerShell V4

Ejemplo 1: Validación del contenido de la plantilla especificada. El resultado detalla las capacidades, la descripción y los parámetros de la plantilla.

```
Test-CFNTemplate -TemplateBody "{TEMPLATE CONTENT HERE}"
```

Ejemplo 2: Validación de la plantilla especificada a la que se ha accedido con una URL de Amazon S. El resultado detalla las capacidades, la descripción y los parámetros de la plantilla.

```
Test-CFNTemplate -TemplateURL https://s3.amazonaws.com/amzn-s3-demo-bucket/
templatefile.template
```

- Para obtener información sobre la API, consulte [ValidateTemplate](#) en Herramientas de AWS para PowerShell Cmdlet Reference (V4).

Herramientas para PowerShell V5

Ejemplo 1: Validación del contenido de la plantilla especificada. El resultado detalla las capacidades, la descripción y los parámetros de la plantilla.

```
Test-CFNTemplate -TemplateBody "{TEMPLATE CONTENT HERE}"
```

Ejemplo 2: Validación de la plantilla especificada a la que se ha accedido con una URL de Amazon S3. El resultado detalla las capacidades, la descripción y los parámetros de la plantilla.

```
Test-CFNTemplate -TemplateURL https://s3.amazonaws.com/amzn-s3-demo-bucket/
templatefile.template
```

- Para obtener información sobre la API, consulte [ValidateTemplate](#) en Herramientas de AWS para PowerShell Cmdlet Reference (V5).

A continuación, se muestra un ejemplo de respuesta que produce un error de validación.

```
{
  "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

El comando `validate-template` está diseñado para comprobar únicamente la sintaxis de la plantilla. No garantiza que los valores de propiedad que ha especificado para un recurso sean válidos para ese recurso. Tampoco determina la cantidad de recursos que existirán cuando se cree la pila.

Para comprobar la validez operativa, debe intentar crear la pila. No hay entorno aislado ni área de pruebas para pilas de CloudFormation, por lo que se le cobrará por los recursos que cree durante las pruebas.

Carga de artefactos locales en un bucket de S3 con la AWS CLI

Puede utilizar la AWS CLI para cargar artefactos locales a los que se hace referencia en una plantilla de CloudFormation a un bucket de Amazon S3. Los artefactos locales son archivos a los que hace referencia en su plantilla. En lugar de cargar de forma manual los archivos en un bucket de S3 y, a continuación, añadir la ubicación a su plantilla, puede especificar artefactos locales en su plantilla y luego usar el comando [package](#) para cargarlos rápidamente.

Un artefacto local es una ruta a un archivo o carpeta que el comando `package` carga en Amazon S3. Por ejemplo, un artefacto puede ser una ruta local al código fuente de su función de AWS Lambda o un archivo OpenAPI de la API de REST de Amazon API Gateway.

Al usar el comando `package`:

- Si especifica un archivo, el comando se carga directamente en el bucket de S3.
- Si especifica una carpeta, el comando crea un archivo `.zip` para la carpeta y, a continuación, carga el archivo `.zip`.
- Si no especifica una ruta, el comando crea un archivo `.zip` para el directorio de trabajo y, a continuación, lo carga.

Puede especificar una ruta absoluta o relativa, donde la ruta relativa es relativa respecto a la ubicación de la plantilla.

Después de cargar los artefactos, el comando devuelve una copia de la plantilla, sustituyendo las referencias a artefactos locales con la ubicación de S3 donde el comando cargó los artefactos. A continuación, puede utilizar la plantilla devuelta para crear o actualizar una pila.

Note

Solo puede usar artefactos locales para propiedades de recursos compatibles con el comando `package`. Para obtener más información acerca de este comando y una lista de las propiedades de recursos compatibles, consulte la documentación de [package](#) en la [Referencia de comandos de AWS CLI](#).

Requisitos previos

Antes de comenzar, debe tener un bucket de Amazon S3 existente.

Empaquetar e implementar una plantilla con artefactos locales

La siguiente plantilla especifica el artefacto local del código fuente de una función de Lambda. El código fuente se almacena en la carpeta `/home/user/code/lambdafunction`.

Plantilla original

```
{
  "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"
      }
    }
  }
}
```

El siguiente comando [package](#) crea y carga un archivo `.zip` de la carpeta del código de origen de la función en la raíz del bucket especificado.

```
aws cloudformation package \  
  --s3-bucket amzn-s3-demo-bucket \  
  --template /path_to_template/template.json \  
  --output-template-file packaged-template.json \  
  --output json
```

El comando genera una nueva plantilla en la ruta especificada por `--output-template-file`. Sustituye la referencia del artefacto con la ubicación de Amazon S3, tal y como se muestra a continuación. El archivo `.zip` se nombra utilizando la suma de comprobación MD5 del contenido de la carpeta, en lugar de utilizar el nombre de la carpeta en sí.

Plantilla resultante

```
{  
  "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"  
      }  
    }  
  }  
}
```

Después de empaquetar los artefactos de su plantilla, implemente la plantilla procesada mediante el comando [deploy](#).

```
aws cloudformation deploy \  
  --template-file packaged-template.json \  
  --stack-name stack-name
```

Al implementar plantillas de más de 51 200 bytes, utilice el comando `deploy` con la opción `--s3-bucket` para cargar su plantilla en S3, como en el siguiente ejemplo.

```
aws cloudformation deploy \  
  --template-file packaged-template.json \  
  --s3-bucket amzn-s3-demo-bucket
```

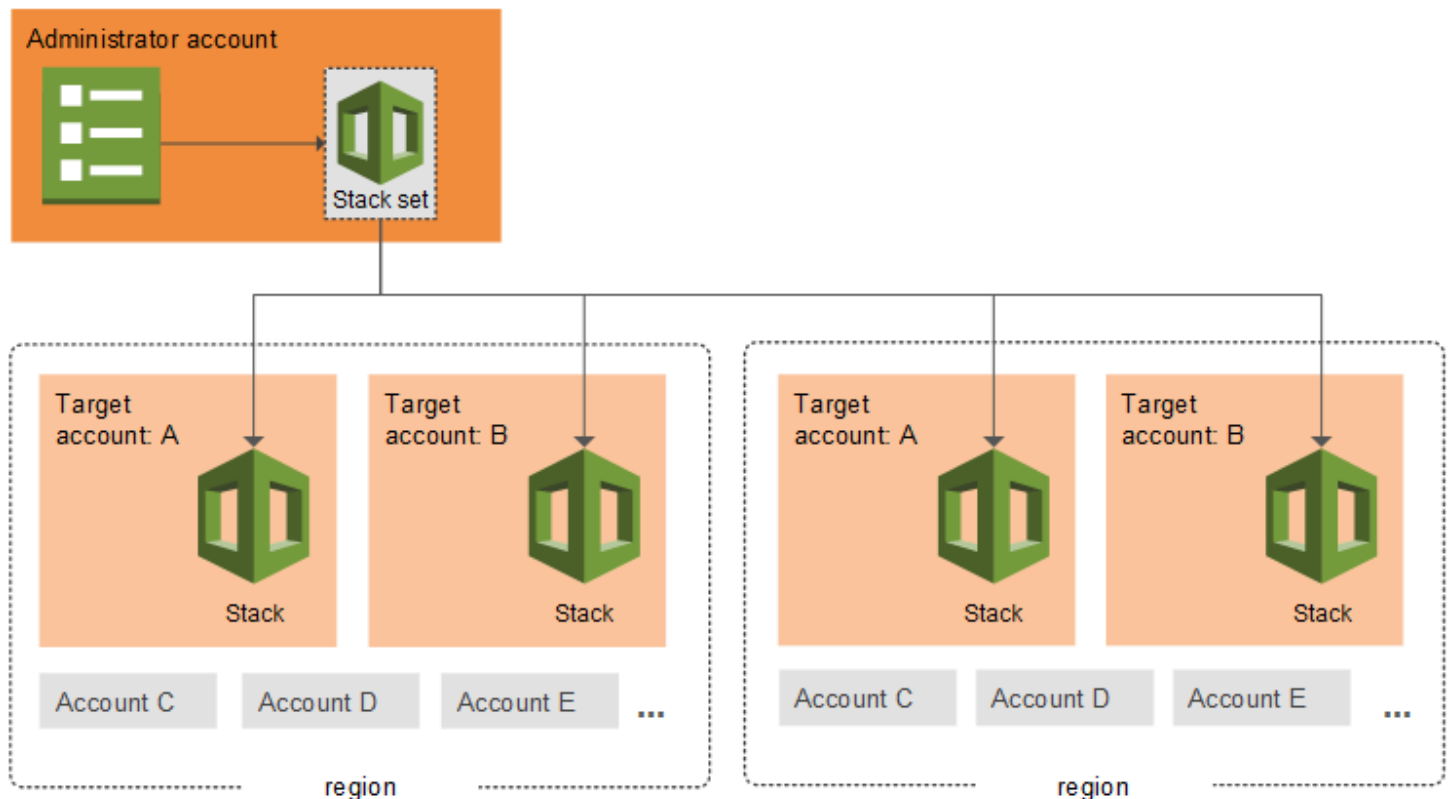
```
--stack-name stack-name \  
--s3-bucket amzn-s3-demo-bucket
```

 Note

Para ver otro ejemplo del uso del comando `package` para cargar artefactos locales, consulte [División de una plantilla en piezas reutilizables utilizando pilas anidadas](#).

Administración de pilas entre cuentas y regiones con StackSets

CloudFormation StackSets amplía la funcionalidad de las pilas al permitirle crear, actualizar o eliminar pilas de varias cuentas y Regiones de AWS con una sola operación. Con una cuenta de administrador, defina y gestione una plantilla de CloudFormation y utilice esta plantilla como base para aprovisionar pilas en cuentas de destino seleccionadas a lo largo de las Regiones de AWS especificadas.



En esta sección se incluye información que lo ayuda a comenzar a utilizar StackSets y se responde a preguntas comunes acerca de cómo trabajar con StackSets y resolver los problemas de creación, actualización y eliminación de los StackSets.

Temas

- [Conceptos de StackSets](#)
- [Requisitos previos para utilizar CloudFormation StackSets](#)
- [Introducción a los StackSets con una plantilla de ejemplo](#)
- [Creación de StackSets de CloudFormation con permisos autoadministrados](#)

- [Creación de StackSets de CloudFormation con permisos administrados por servicios](#)
- [Habilitar o deshabilitar las implementaciones automáticas para StackSets en AWS Organizations](#)
- [Actualización de StackSets de CloudFormation](#)
- [Adición pilas a CloudFormation StackSets](#)
- [Anulación de los valores de los parámetros en pilas en el StackSet de CloudFormation](#)
- [Eliminación de pilas de CloudFormation StackSets](#)
- [Eliminación de StackSets de CloudFormation](#)
- [Cómo evitar las implementaciones fallidas de StackSets mediante puertas de cuentas de destino](#)
- [Elección del modo de simultaneidad para CloudFormation StackSets](#)
- [Detección de desviaciones en CloudFormation StackSets](#)
- [Importación de pilas en CloudFormation StackSets](#)
- [Prácticas recomendadas para utilizar CloudFormation StackSets](#)
- [Plantilla de ejemplo de CloudFormation StackSets](#)
- [Solución de problemas de CloudFormation StackSets](#)

Conceptos de StackSets

Los siguientes conceptos y terminología son fundamentales para entender y utilizar StackSets.

Temas

- [Cuentas de administrador y de destino](#)
- [CloudFormation StackSets](#)
- [Modelos de permisos para StackSets](#)
- [Instancias de pila](#)
- [Operaciones de StackSets](#)
- [Opciones de operaciones con StackSets](#)
- [Etiquetas](#)
- [Códigos de estado de StackSets](#)
- [Códigos de estado de la instancia de pila](#)

Cuentas de administrador y de destino

Una cuenta de administrador es la Cuenta de AWS en la que se crean los StackSets. Para StackSets con permisos administrados por servicios, la cuenta de administrador es la cuenta de administración de la organización o una cuenta de administrador delegado. Para administrar un StackSet, inicie sesión en la cuenta de administrador de AWS que creó el StackSet.

Una cuenta de destino es la cuenta en la que se crean, actualizan o eliminan una o varias pilas del StackSet. Para poder utilizar un StackSet para crear pilas en una cuenta de destino, configure una relación de confianza entre las cuentas de administrador y de destino.

CloudFormation StackSets

Un StackSet sirve como contenedor para múltiples pilas que se implementan en cuentas de Cuentas de AWS y regiones específicas. Cada pila se basa en la misma plantilla de CloudFormation, pero puede personalizar pilas individuales mediante parámetros.

Una vez definido un StackSet, puede crear, actualizar o eliminar pilas de las cuentas de destino y las Regiones de AWS que especifique. Al crear, actualizar o eliminar pilas, también puede especificar las preferencias de operación. Por ejemplo, incluya el orden de las regiones en las que desea llevar a cabo la operación, el umbral de tolerancia a errores antes de que se detengan las operaciones de pila y el número de cuentas que llevan a cabo operaciones de pila simultáneamente.

Un StackSet es un recurso regional. Si crea un StackSet configurado en una Región de AWS, solo puede verla o cambiarla cuando vea esa región.

Modelos de permisos para StackSets

Puede crear StackSets mediante permisos autoadministrados o permisos administrados por servicios.

Con permisos autoadministrados puede crear los roles de IAM que StackSets necesita para la implementación en cuentas y regiones. Estos roles son necesarios para establecer una relación de confianza entre la cuenta desde la que está administrando el StackSet y la cuenta en la que está implementando instancias de pila. Con este modelo de permisos, StackSets puede implementar en cualquier Cuenta de AWS en la que tenga permisos para crear un rol de IAM.

Con los permisos administrados por servicios puede implementar instancias de pilas en cuentas administradas por AWS Organizations. Al usar el modelo de permisos, no tiene que crear los roles

de IAM necesarios; StackSets crea los roles de IAM en su nombre. Con este modelo, también puede activar implementaciones automáticas en cuentas que agregará a su organización en el futuro.

AWS Organizations se integra con CloudFormation y ayuda a administrar y gobernar su entorno de forma centralizada a medida que escala y hace crecer sus recursos de AWS.

- Cuenta de administración: la cuenta que usa para crear la organización. Para obtener más información, consulte [Terminología y conceptos de AWS Organizations](#).
- Administrador delegado: un servicio de AWS compatible puede registrar una cuenta miembro de AWS en la organización como administrador de las cuentas de la organización en ese servicio. Para obtener más información, consulte [Servicios de AWS que se pueden utilizar con AWS Organizations](#).

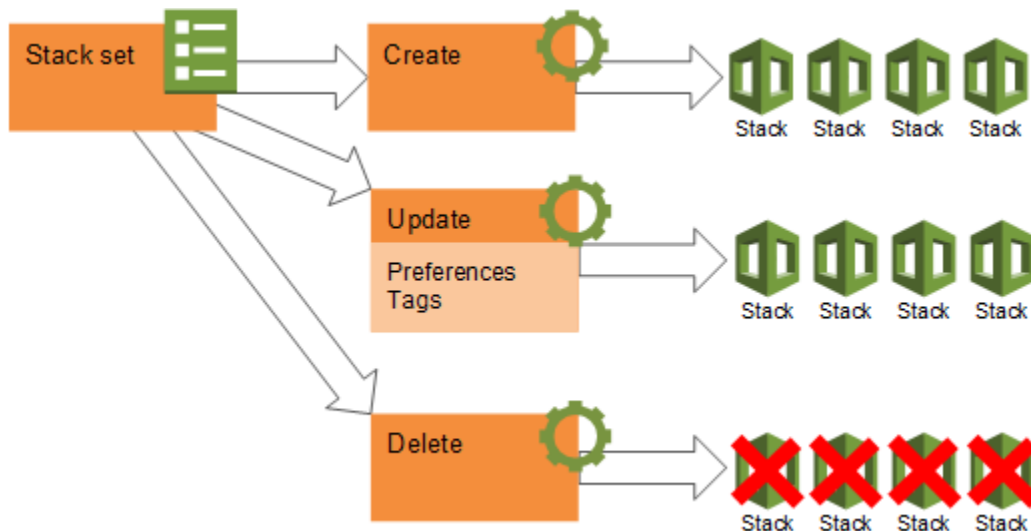
Para obtener más información acerca de cómo crear y administrar StackSets con permisos administrados por servicios, consulte los temas siguientes:

- [Activación del acceso de confianza para StackSets con AWS Organizations](#)
- [Registro de una cuenta miembro como administrador delegado](#)
- [Creación de StackSets de CloudFormation con permisos administrados por servicios](#)

Instancias de pila

Una instancia de pila es una referencia a una pila en una cuenta de destino dentro de una región. Puede existir una instancia de pila sin una pila. Por ejemplo, si la pila no se pudo crear por alguna razón, la instancia de pila muestra el motivo del error de creación de la pila. Una instancia de pila se asocia con un solo StackSet.

En la siguiente figura se muestran las relaciones lógicas entre los StackSets, las operaciones de pila y las pilas. Al actualizar un StackSet, todas las instancias de pila asociadas se actualizan en todas las cuentas y regiones.



Operaciones de StackSets

Puede llevar a cabo las siguientes operaciones en StackSets.

Creación de un StackSet

Para crear un nuevo StackSet es necesario especificar una plantilla de CloudFormation que desee utilizar para crear pilas, especificar las cuentas de destino en las que desee crear pilas e identificar las Regiones de AWS en las que desee implementar pilas en sus cuentas de destino. Un StackSet garantiza la implementación coherente de los mismos recursos de pila, con la misma configuración, en todas las cuentas de destino especificadas dentro de las regiones que elija.

Actualización de un StackSet

Al actualizar un StackSet, envía cambios a las pilas del StackSet. Puede actualizar un StackSet de cualquiera de las siguientes formas. Las actualizaciones de la plantilla siempre afectan a todas las pilas; no puede actualizar la plantilla de manera selectiva para algunas pilas del StackSet, pero no para otras.

- Cambiar la configuración existente en la plantilla o añadir nuevos recursos, como actualizar la configuración de los parámetros de un servicio específico o añadir nuevas instancias Amazon EC2.
- Reemplazar la plantilla por otra.
- Añadir pilas en cuentas de destino adicionales o existentes, en regiones adicionales o existentes.

Eliminar pilas

Al eliminar pilas, elimina una pila y todos sus recursos asociados de las cuentas de destino que especifique, dentro de las regiones que indique. Puede eliminar pilas de las siguientes maneras.

- Eliminar pilas de algunas cuentas de destino, mientras deja en ejecución otras pilas en otras cuentas de destino.
- Eliminar pilas de algunas regiones, dejando en ejecución pilas en otras regiones.
- Eliminar pilas del StackSet, pero guardarlas para que sigan ejecutándose independientemente del StackSet al elegir la opción Retener pilas. Luego puede administrar las pilas retenidas fuera del StackSet en CloudFormation.
- Eliminar todas las pilas del StackSet como preparación para eliminar todo el StackSet.

Eliminación de un StackSet

Solo puede eliminar el StackSet cuando no haya instancias de pila en él.

Opciones de operaciones con StackSets

Las opciones que se describen en esta sección ayudan a controlar el tiempo y el número de errores que se permiten para llevar a cabo correctamente operaciones con StackSets y evitan que pierda recursos de pila.

Maximum concurrent accounts

Esta configuración, disponible al crear, actualizar y eliminar flujos de trabajo, permite especificar el número máximo o el porcentaje de cuentas de destino en el que se lleva a cabo una operación en un momento dado. Un número o porcentaje menor significa que una operación se lleva a cabo en menos cuentas de destino a la vez. Las operaciones se llevan a cabo en una región a la vez, en el orden especificado en el cuadro Deployment order (Orden de implementación). Por ejemplo, si va a implementar pilas en 10 cuentas de destino en dos regiones, al establecer Maximum concurrent accounts (Número máximo de cuentas simultáneas) en 50 y By percentage (Por porcentaje), se implementarán pilas en cinco cuentas de la primera región, y luego en las segundas cinco cuentas de la primera región, antes de pasar a la siguiente región y empezar la implementación en las primeras cinco cuentas de destino.

Cuando elige By percentage (Por porcentaje), si el porcentaje especificado no representa un número entero de sus cuentas especificadas, CloudFormation lo redondea hacia abajo. Por ejemplo, si va a implementar pilas en 10 cuentas de destino, y establece Maximum concurrent

accounts (Número máximo de cuentas simultáneas) en 25 y By percentage (Por porcentaje), CloudFormation hace un redondeo hacia abajo y pasa de implementar 2,5 pilas de forma simultánea (lo que no sería posible) a implementar dos pilas simultáneamente.

Tenga en cuenta que esta configuración le permite especificar el número máximo de operaciones. En implementaciones grandes, el número real de cuentas sobre las que se actúa simultáneamente puede ser menor en determinadas circunstancias debido a la limitación controlada de los servicios.

El número máximo de cuentas simultáneas puede depender del valor de la tolerancia a errores en función del modo de simultaneidad. Si el modo de simultaneidad está configurado con una tolerancia a errores estricta, el número máximo de cuentas simultáneas puede ser como máximo una vez más que el valor de tolerancia a errores.

Modo de simultaneidad

Esta configuración, disponible en los flujos de trabajo de creación, actualización y eliminación, le permite elegir cómo se comporta el nivel de simultaneidad durante las operaciones con StackSets. Para obtener más información, consulte [Elección del modo de simultaneidad para CloudFormation StackSets](#).

Failure tolerance

Esta configuración, disponible al crear, actualizar y eliminar flujos de trabajo, permite especificar el número o el porcentaje máximo de fallos en las operaciones de pila que pueden producirse, por región, a partir del cual CloudFormation detiene una operación automáticamente. Un número o porcentaje menor significa que la operación se lleva a cabo en menos pilas, pero puede empezar antes a solucionar problemas de las operaciones con errores. Por ejemplo, si va a actualizar 10 pilas en 10 cuentas de destino de tres regiones, establecer Tolerancia a errores en 20 y Por porcentaje significa que pueden fracasar como máximo dos actualizaciones de pila en una región para que la operación pueda continuar. Si se produce un error en una tercera pila de la misma región, CloudFormation detiene la operación. Si una pila no se puede actualizar en la primera región, la operación de actualización continúa en esa región y luego pasa a la siguiente región. Si dos pilas no se pueden actualizar en la segunda región, se alcanza la tolerancia a errores del 20 %; si se produce un error en una tercera pila de la región, CloudFormation detiene la operación de actualización y no continúa en las regiones siguientes.

Cuando elige By percentage (Por porcentaje), si el porcentaje especificado no representa un número entero de sus pilas dentro de cada región, CloudFormation lo redondea hacia abajo. Por ejemplo, si va a implementar pilas en 10 cuentas de destino de tres regiones, y establece

Tolerancia a errores en 25 y Por porcentaje, CloudFormation hace un redondeo hacia abajo y pasa de una tolerancia a errores de 2,5 pilas (lo que no sería posible) a una tolerancia a errores de dos pilas por región.

Retain stacks

Esta configuración, disponible en los flujos de trabajo de eliminación de pilas, le permite mantener en ejecución las pilas y sus recursos incluso después eliminarlos de un StackSet. Cuando retiene las pilas, CloudFormation deja intactas las pilas de las cuentas y regiones individuales. Las pilas se desasocian del StackSet, pero la pila y sus recursos se guardan. Cuando la operación de eliminación de pilas ha finalizado, administra las pilas retenidas en CloudFormation, en la cuenta de destino (no en la cuenta de administrador) en la que se crearon.

Simultaneidad de región

Esta configuración, disponible en los flujos de trabajo de creación, actualización y eliminación, le permite elegir cómo se implementa StackSets en las regiones.

Secuencial: operación de implementación de StackSets en una región a la vez, tal como se especifica en el cuadro de Deployment order (Orden de implementación) de región, siempre y cuando los errores de implementación de una región no superen una tolerancia de error especificada. La implementación secuencial es la selección predeterminada.

Paralela: operaciones de implementación de StackSets en todas las regiones especificadas simultáneamente, siempre y cuando los errores de implementación de una región no superen una tolerancia a errores especificada.

Etiquetas

Puede agregar etiquetas durante las operaciones de creación y actualización de StackSets especificando pares de clave y valor. Las etiquetas son útiles para ordenar y filtrar los recursos de los StackSets de cara a la facturación y la asignación de costos. Para obtener más información acerca del uso de etiquetas en AWS, consulte [Organización y rastreo de costos mediante las etiquetas de asignación de costos de AWS](#) en la Guía del usuario de Administración de facturación y costos de AWS. Después de especificar el par de clave-valor, elija + para guardar la etiqueta. Puede eliminar las etiquetas que ya no esté utilice con la X roja a la derecha de una etiqueta.

Las etiquetas que se aplican a los StackSets se aplican a todas las pilas y a los recursos creados por sus pilas. Las etiquetas se pueden agregar solo en la pila en CloudFormation, pero puede que esas etiquetas no se muestren en StackSets.

Aunque StackSets no agrega actualmente etiquetas definidas por el sistema, no debe usar la cadena `aws:` al principio de los nombres de clave de ninguna etiqueta.

Códigos de estado de StackSets

CloudFormation StackSets genera códigos de estado para las operaciones con StackSets.

En la siguiente tabla se describen los códigos de estado para las operaciones con StackSets.

RUNNING

La operación está en curso actualmente.

SUCCEEDED

La operación terminó sin superar la tolerancia a fallos de la operación.

FAILED

El número de pilas en las que la operación no se completó superó la tolerancia a errores definida por el usuario. El valor de tolerancia a errores que ha establecido para una operación aplica a cada región durante las operaciones de creación y actualización de pilas. Si el número de pilas con errores dentro de una región supera la tolerancia a errores, el estado de la operación en la región cambia a `FAILED`. El estado de la operación en su conjunto también se establece en `FAILED` y CloudFormation cancela la operación en las regiones restantes.

QUEUED

[Service-managed permissions] Para implementaciones automáticas que requieren una secuencia de operaciones, la operación entra en una cola para llevarla a cabo. Por ejemplo:

- Si se mueve una cuenta de una unidad organizativa (OU), OU1, a otra, OU2, se activa una implementación automática. StackSets ejecuta una operación de eliminación para quitar la instancia de pila de la cuenta de OU1 de destino en la región de destino y pone en cola una operación de creación para añadir una instancia de pila a la cuenta de OU2 de destino en la región de destino.
- Si se añade una cuenta AccountA a una unidad organizativa, se activa una implementación automática. StackSets ejecuta una operación de creación para añadir una instancia de pila a AccountA en la región de destino. Si añade otra cuenta AccountB a la unidad organizativa mientras se ejecuta la operación de creación, StackSets pone en cola una segunda operación de creación. Cuando se completa la primera operación de creación, StackSets ejecuta la

segunda operación de creación para añadir una instancia de pila a AccountB en la región de destino.

STOPPING

La operación se encuentra en proceso de detención, a petición del usuario.

STOPPED

La operación se detuvo a petición del usuario.

Códigos de estado de la instancia de pila

CloudFormation StackSets genera códigos de estado para las instancias de pilas.

En la siguiente tabla, se describen los códigos de estado para las instancias de pilas dentro de los StackSets.

CURRENT

La pila está actualizada con el StackSet.

OUTDATED

La pila no está actualizada con el StackSet por una de las siguientes razones.

- Se produjo un error en una operación [CreateStackSet](#) o [UpdateStackSet](#) en la pila asociada.
- La pila formaba parte de una operación [CreateStackSet](#) o [UpdateStackSet](#) que falló, o se detuvo antes de crear o actualizar la pila.

INOPERABLE

Se produjo un error en una operación [DeleteStackInstances](#) y dejó la pila en un estado inestable. Las pilas que se encuentran en este estado quedan excluidas de otras operaciones [UpdateStackSet](#). Es posible que necesite realizar una operación [DeleteStackInstances](#), con `RetainStacks` establecido en `true`, para eliminar la instancia de la pila y, a continuación, tenga que eliminar la pila manualmente.

CANCELLED

La operación de la cuenta y la región especificadas se ha cancelado. Esto se debe a que un usuario detuvo la operación del StackSet o a que se excedió la tolerancia a errores de la operación con el StackSet.

FAILED

Ha habido un error en la región y cuenta especificadas. Si la operación con el StackSet falla en un número suficiente de cuentas de una misma región, se puede exceder la tolerancia a errores del total de la operación con el StackSet.

FAILED_IMPORT

Se produjo un error en la importación de la instancia de pila en la cuenta y región especificadas y dejó la pila en un estado inestable. Una vez que se hayan solucionado los problemas que causaron el error, se puede volver a intentar la operación de importación. Si la cantidad necesaria de operaciones con el StackSet falla en un número suficiente de cuentas de una misma región, se puede exceder la tolerancia a errores del total de la operación con el StackSet.

PENDING

La operación aún no ha empezado en la cuenta y región especificadas.

RUNNING

La operación está actualmente en curso en la cuenta y región especificadas.

SKIPPED_SUSPENDED_ACCOUNT

No se realizó la operación en la cuenta y región especificadas porque la cuenta estaba suspendida en el momento de la operación.

SUCCEEDED

La operación se ha completado correctamente en la cuenta y región especificadas.

Requisitos previos para utilizar CloudFormation StackSets

StackSets amplía la funcionalidad de las pilas para que pueda crear, actualizar o eliminar pilas de varias cuentas y regiones con una sola operación.

Dado que StackSets lleva a cabo operaciones en varias cuentas, antes de que pueda empezar a crear su primer StackSet debe tener los permisos necesarios definidos en su Cuentas de AWS.

Puede administrar StackSets mediante permisos autoadministrados o permisos administrados por servicios.

- En el caso de los StackSets autoadministrados, debe crear y gestionar los roles de IAM en cada cuenta de destino y Región de AWS. Para obtener más información, consulte [Concesión de permisos autoadministrados](#).
- En el caso de los StackSets administrados por servicios, no necesita crear y gestionar manualmente los roles de IAM en cada cuenta; AWS se encarga de la creación del rol y los permisos por usted. Para obtener más información, consulte [Activación del acceso de confianza](#).

Temas

- [Preparación para llevar a cabo operaciones con StackSets en Regiones de AWS que están deshabilitadas de forma predeterminada](#)
- [Concesión de permisos autoadministrados](#)
- [Activación del acceso de confianza para StackSets con AWS Organizations](#)

Preparación para llevar a cabo operaciones con StackSets en Regiones de AWS que están deshabilitadas de forma predeterminada

Las Regiones de AWS introducidas después del 20 de marzo de 2019, como Asia-Pacífico (Hong Kong), están desactivadas de forma predeterminada. Debe habilitar estas regiones para sus cuentas antes de poder usarlas. Para obtener más información, consulte [Habilitar o deshabilitar Regiones de AWS en su cuenta](#) en la Guía de referencia de AWS Account Management.

Para crear un StackSet a partir de la cuenta de administrador de un StackSet (si se utilizan permisos autoadministrados) o de la cuenta de administración de la organización (si se utilizan permisos administrados por servicios) en una región que está deshabilitada de forma predeterminada, primero debe habilitar esa región para la cuenta de administrador o administración.

Para que CloudFormation cree o actualice correctamente una instancia de pila:

- La cuenta de destino debe residir en una región que esté habilitada actualmente para esa cuenta de destino.
- La cuenta de administrador del StackSet o la cuenta de administración de la organización debe tener la misma región habilitada que la cuenta de destino.

 Important

Tenga en cuenta que, durante las operaciones con StackSets, las cuentas de administrador y de destino intercambian metadatos relacionados con las propias cuentas, además del StackSet y de las instancias del StackSet involucradas.

Además, si deshabilita una región que contiene una cuenta en la que residen instancias de un StackSet, será responsable de eliminar dichas instancias o recursos, si lo desea. Además, tenga en cuenta que los metadatos relativos a la cuenta de destino en la región desactivada se conservarán en la cuenta de administrador.

Concesión de permisos autoadministrados

En este tema se proporcionan instrucciones sobre cómo crear los roles de servicio de IAM que requieren los StackSets para implementarse en cuentas y Regiones de AWS con permisos autoadministrados. Estos roles son necesarios para establecer una relación de confianza entre la cuenta desde la que está administrando el StackSet y la cuenta en la que está implementando instancias de pila. Con este modelo de permisos, StackSets puede implementar en cualquier Cuenta de AWS en la que tenga permisos para crear un rol de IAM.

Para usar los permisos administrados por servicios, consulte en su lugar [Activación del acceso de confianza](#).

Temas

- [Descripción general de los permisos autoadministrados](#)
- [Concesión de permisos a todos los usuarios de la cuenta de administrador para administrar pilas en todas las cuentas de destino](#)
- [Configuración de opciones de permisos avanzados para operaciones con StackSet](#)
- [Configuración de claves globales para mitigar problemas de adjuntos confusos](#)

Descripción general de los permisos autoadministrados

Antes de crear un StackSet con permisos autoadministrados, debe haber creado roles de servicio de IAM en cada cuenta.

A continuación, indicamos los pasos básicos:

1. Determine cuál Cuenta de AWS es la cuenta de administrador.

Los StackSet se crean en esta cuenta de administrador. Una cuenta de destino es la cuenta en la que se crean pilas individuales que pertenecen a un StackSet.

2. Determine cómo desea estructurar los permisos para el StackSet.

La configuración de permisos más sencilla (y más permisiva) es aquella en que se proporciona a todos los usuarios y grupos de la cuenta de administrador la capacidad de crear y actualizar todos los StackSets administrados a través de dicha cuenta. Si necesita de un control más preciso, puede configurar permisos para especificar lo siguiente:

- Qué usuarios y grupos pueden llevar a cabo operaciones con StackSets en qué cuentas de destino.
- Qué recursos pueden incluir los usuarios y grupos en sus StackSets.
- Qué operaciones con StackSets pueden llevar a cabo usuarios y grupos concretos.

3. Cree los roles de servicio de IAM necesarios en las cuentas de administrador y de destino para definir los permisos que desee.

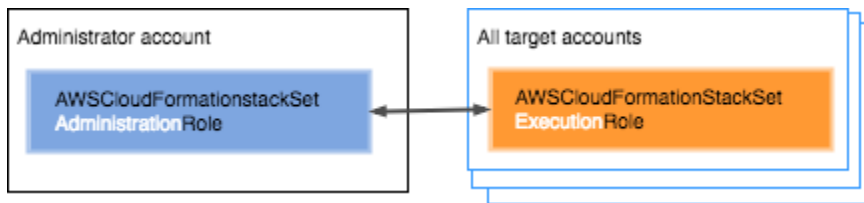
En concreto, los dos roles obligatorios son los siguientes:

- `AWSCloudFormationStackSetAdministrationRole`: este rol se implementa en la cuenta de administrador.
- `AWSCloudFormationStackSetExecutionRole`: este rol se implementa en todas las cuentas en las que se crean instancias de pila.

Concesión de permisos a todos los usuarios de la cuenta de administrador para administrar pilas en todas las cuentas de destino

En esta sección se muestra cómo configurar permisos para permitir que todos los usuarios y grupos de la cuenta de administrador lleven a cabo operaciones con StackSets en todas las cuentas de destino. Se le guía en la creación de los roles de servicio de IAM necesarios en las cuentas de administrador y de destino. Cualquiera en la cuenta de administrador podrá crear, actualizar o eliminar cualquier pila en cualquiera de las cuentas de destino.

Al estructurar los permisos de esta manera, los usuarios no transfieren un rol de administración al crear o actualizar un StackSet.



Administrator account

En la cuenta de administrador, cree un rol de IAM denominado `AWSCloudFormationStackSetAdministrationRole`.

Para ello, cree una pila a partir de la siguiente plantilla de CloudFormation, disponible en <https://s3.amazonaws.com/cloudformation-sample-templates-us-east-1/AWSCloudFormationStackSetAdministrationRole.yml>.

Example Ejemplo de política de permisos

El rol de administración creado mediante la plantilla anterior incluye la siguiente política de permisos.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "sts:AssumeRole"
      ],
      "Resource": [
        "arn:aws:iam::*:role/AWSCloudFormationStackSetExecutionRole"
      ],
      "Effect": "Allow"
    }
  ]
}
```

Example Política de confianza de ejemplo 1

La plantilla anterior también incluye la siguiente política de confianza que otorga al servicio permiso para utilizar el rol de administración y los permisos asociados al rol.

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

Example Política de confianza de ejemplo 2

Para implementar instancias de pila en una cuenta de destino que resida en una región deshabilitada de forma predeterminada, también tendrá que incluir la entidad principal de servicio regional para esa región. Cada región que esté deshabilitada de forma predeterminada tendrá su propia entidad principal de servicio regional.

En el siguiente ejemplo, la política de confianza otorga al servicio permiso para utilizar el rol de administración en la región Asia-Pacífico (Hong Kong) (ap-east-1), una región que está deshabilitada de forma predeterminada.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "cloudformation.amazonaws.com",
          "cloudformation.ap-east-1.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Para obtener más información, consulte [Preparación para llevar a cabo operaciones con StackSets en Regiones de AWS que están deshabilitadas de forma predeterminada](#). Para obtener una lista de códigos de región, consulte [Puntos de conexión regionales](#) en la Guía de Referencia general de AWS.

Target accounts

En cada cuenta de destino, cree un rol de servicio denominado `AWSCloudFormationStackSetExecutionRole` que confíe en la cuenta de administrador. El rol debe tener este nombre exacto. Para ello, cree una pila a partir de la siguiente plantilla de CloudFormation, disponible en <https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/AWSCloudFormationStackSetExecutionRole.yml>. Cuando utiliza esta plantilla, se le solicita que proporcione el ID de la cuenta de administrador con la que su cuenta de destino debe tener una relación de confianza.

Important

Tenga en cuenta que esta plantilla concede acceso de administrador. Después de utilizar la plantilla para crear un rol de ejecución de la cuenta de destino, debe examinar los permisos de la declaración de política para los tipos de recursos que están creando mediante StackSets.

El rol de servicio de la cuenta de destino requiere permisos para realizar las operaciones especificadas en la plantilla de CloudFormation. Por ejemplo, si su plantilla está creando un bucket de S3, necesita permisos para crear nuevos objetos de S3. Su cuenta de destino siempre necesita permisos de CloudFormation completos, que incluyen permisos para crear, actualizar, eliminar y describir pilas.

Example Ejemplo 1 de política de permisos

El rol creado con esta plantilla permite la siguiente política en una cuenta de destino.

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



```
        "Action": "*",
        "Resource": "*"
    }
]
}
```

Example Ejemplo 2 de política de permisos

El siguiente ejemplo muestra la declaración de política con los permisos mínimos para que StackSets funcione. Para crear pilas en las cuentas de destino que utilicen los recursos de servicios que no sean CloudFormation, debe agregar las acciones del servicio y los recursos a la instrucción de política `AWSCloudFormationStackSetExecutionRole` de cada cuenta de destino.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudformation:*"
      ],
      "Resource": "*"
    }
  ]
}
```

Example Política de confianza de ejemplo

Con la plantilla se crea la siguiente relación de confianza. El ID de la cuenta del administrador se muestra como *id_cuenta_admin*.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:root"
      },

```

```
        "Action": "sts:AssumeRole"
    }
  ]
}
```

Puede configurar la relación de confianza de un rol de ejecución de la cuenta de destino existente para confiar en un rol específico en la cuenta de administrador. Si elimina el rol en la cuenta de administrador y crea uno nuevo para sustituirlo, debe configurar las relaciones de confianza de la cuenta de destino con el nuevo rol de cuenta de administrador, representada por *id_cuenta_admin* en el ejemplo anterior.

Configuración de opciones de permisos avanzados para operaciones con StackSet

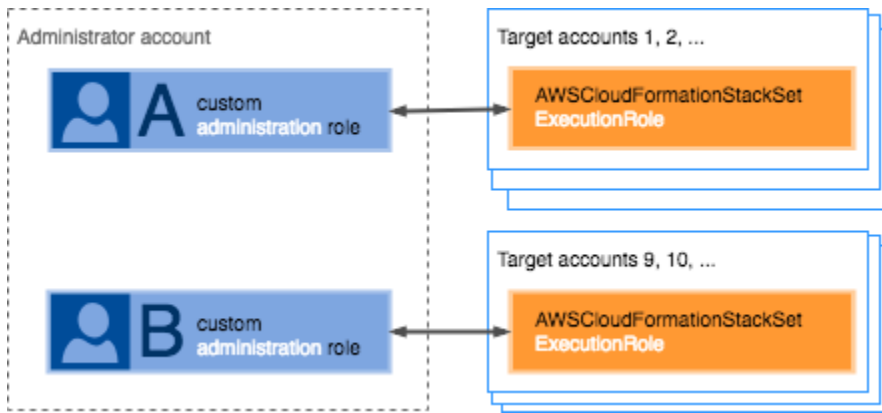
Si necesita un control más preciso de los StackSets que los usuarios y grupos crean a través de una única cuenta de administrador, puede utilizar roles de IAM para especificar lo siguiente:

- Qué usuarios y grupos pueden llevar a cabo operaciones con StackSets en qué cuentas de destino.
- Qué recursos pueden incluir los usuarios y grupos en sus StackSets.
- Qué operaciones con StackSets pueden llevar a cabo usuarios y grupos concretos.

Control de los usuarios que pueden llevar a cabo operaciones con StackSets en cuentas de destino específicas

Utilice roles de administración personalizados para controlar qué usuarios y grupos pueden llevar a cabo operaciones con StackSets y en qué cuentas de destino. Podría desear controlar qué usuarios de la cuenta de administrador pueden llevar a cabo operaciones con StackSets y en qué cuentas de destino. Para ello, debe crear una relación de confianza entre cada cuenta de destino y un rol de administración personalizado específico, en lugar de crear el rol de servicio `AWSCloudFormationStackSetAdministrationRole` en la propia cuenta de administrador. A continuación, active usuarios y grupos específicos para utilizar el rol de administración personalizado al llevar a cabo operaciones con StackSets en una cuenta de destino concreta.

Por ejemplo, puede crear la Función A y la Función B en su cuenta de administrador. Puede dar a la Función A permisos para obtener acceso de la cuenta de destino 1 a la cuenta 8. Puede dar a la Función B permisos para obtener acceso de la cuenta de destino 9 a la cuenta 16.



La configuración de los permisos necesarios implica definir un rol de administración personalizado, crear un rol de servicio para la cuenta de destino y conceder a los usuarios permiso para transferir el rol de administración personalizado al llevar a cabo operaciones con StackSets.

En general, así es cómo funciona una vez que dispone de los permisos necesarios: al crear un StackSet, el usuario debe especificar un rol de administración personalizado. El usuario debe tener permiso para transferir el rol a CloudFormation. Además, el rol de administración personalizado debe tener una relación de confianza con las cuentas de destino especificadas para el StackSet. CloudFormation crea el StackSet y le asocia el rol de administración personalizado. Al actualizar un StackSet, el usuario debe especificar explícitamente un rol de administración personalizado, aunque sea el mismo que se utilizó con este StackSet anteriormente. CloudFormation utiliza ese rol para actualizar la pila, siempre que se cumplan los requisitos anteriores.

Administrator account

Example Ejemplo de política de permisos

Para cada StackSet, cree un rol de administración personalizado con permisos para asumir el rol de ejecución de las cuentas de destino.

El nombre del rol de ejecución de las cuentas de destino debe ser el mismo en todas las cuentas de destino. Si el nombre del rol es `AWSCloudFormationStackSetExecutionRole`, StackSets lo usa automáticamente al crear un StackSet. Si especifica un nombre de rol personalizado, los usuarios deben proporcionar el nombre del rol de ejecución al crear un StackSet.

Cree un [rol de servicio de IAM](#) con un nombre personalizado y la siguiente política de permisos. En los ejemplos siguientes, *custom_execution_role* hace referencia al rol de ejecución en las cuentas de destino.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "sts:AssumeRole"
      ],
      "Resource": [
        "arn:aws:iam::111122223333:role/custom_execution_role"
      ],
      "Effect": "Allow"
    }
  ]
}
```

Para especificar varias cuentas en una única instrucción, sepárelas con comas.

```
"Resource": [
  "arn:aws:iam::target_account_id_1:role/custom_execution_role",
  "arn:aws:iam::target_account_id_2:role/custom_execution_role"
]
```

Puede especificar todas las cuentas de destino utilizando un comodín (*) en lugar de un ID de cuenta.

```
"Resource": [
  "arn:aws:iam::*:role/custom_execution_role"
]
```

Example Política de confianza de ejemplo 1

Debe proporcionar una política de confianza para el rol de servicio a fin de definir qué entidades principales de IAM pueden asumir el rol.

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

```
    "Principal": {
      "Service": "cloudformation.amazonaws.com"
    },
    "Action": "sts:AssumeRole"
  }
]
```

Example Política de confianza de ejemplo 2

Para implementar instancias de pila en una cuenta de destino que resida en una región deshabilitada de forma predeterminada, también tendrá que incluir la entidad principal de servicio regional para esa región. Cada región que esté deshabilitada de forma predeterminada tendrá su propia entidad principal de servicio regional.

En el siguiente ejemplo, la política de confianza otorga al servicio permiso para utilizar el rol de administración en la región Asia-Pacífico (Hong Kong) (ap-east-1), una región que está deshabilitada de forma predeterminada.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "cloudformation.amazonaws.com",
          "cloudformation.ap-east-1.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Para obtener más información, consulte [Preparación para llevar a cabo operaciones con StackSets en Regiones de AWS que están deshabilitadas de forma predeterminada](#). Para obtener una lista de códigos de región, consulte [Regional endpoints](#) en la Guía de referencia general de AWS.

Example Ejemplo de política de transferencia de roles

También necesita una política de permisos de IAM para los usuarios de IAM que les permita transferir el rol de administración personalizado al llevar a cabo operaciones con StackSets. Para obtener más información, consulte [Concesión de permisos a un usuario para transferir un rol a un servicio de AWS](#).

En el ejemplo siguiente, *customized_admin_role* hace referencia al rol de administración que tiene que transferir el usuario.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iam:GetRole",
        "iam:PassRole"
      ],
      "Resource": "arn:aws:iam::*:role/customized_admin_role"
    }
  ]
}
```

Target accounts

En cada cuenta de destino, cree un rol de servicio que confíe en el rol de administración personalizado que desee utilizar con esta cuenta.

El rol de la cuenta de destino requiere permisos para realizar las operaciones especificadas en la plantilla de CloudFormation. Por ejemplo, si su plantilla está creando un bucket de S3, necesita permisos para crear nuevos objetos en S3. Su cuenta de destino siempre necesita permisos de CloudFormation completos, que incluyen permisos para crear, actualizar, eliminar y describir pilas.

El nombre del rol de las cuentas de destino debe ser el mismo en todas las cuentas de destino. Si el nombre del rol es `AWSCloudFormationStackSetExecutionRole`, StackSets lo usa automáticamente al crear un StackSet. Si especifica un nombre de rol personalizado, los usuarios deben proporcionar el nombre del rol de ejecución al crear un StackSet.

Example Ejemplo de política de permisos

El siguiente ejemplo muestra la declaración de política con los permisos mínimos para que StackSets funcione. Para crear pilas en las cuentas de destino que utilicen los recursos de servicios que no sean CloudFormation, debe agregar las acciones del servicio y los recursos a la política de permisos.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudformation:*"
      ],
      "Resource": "*"
    }
  ]
}
```

Example Política de confianza de ejemplo

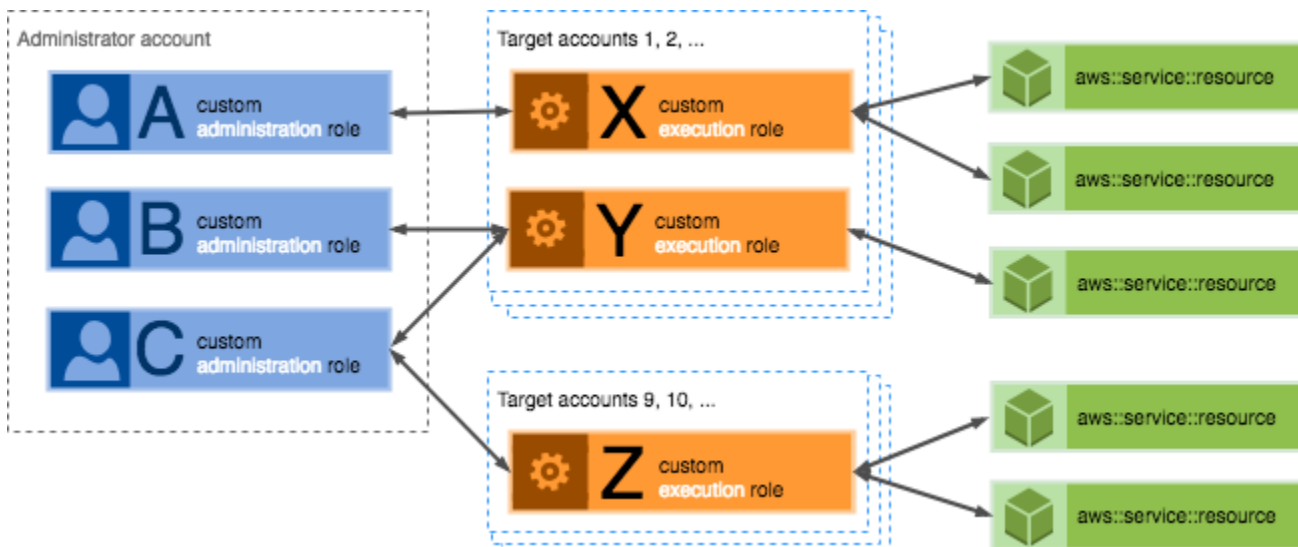
Debe proporcionar la siguiente política de confianza a la hora de crear el rol para definir la relación de confianza.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/customized_admin_role"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Control de los recursos que los usuarios pueden incluir en StackSets específicos

Utilice roles de ejecución personalizados para controlar qué recursos de pila podrán incluir los usuarios y grupos en sus StackSets. Por ejemplo, podría ser interesante configurar un grupo que solo pueda incluir recursos relacionados con Amazon S3 en los StackSets que crean, mientras que otro equipo solo pueda incluir recursos de DynamoDB. Para ello, debe crear una relación de confianza entre el rol de administración personalizado para cada grupo y un rol de ejecución personalizado para cada conjunto de recursos. El rol de ejecución personalizado define qué recursos de pila se pueden incluir en los StackSets. El rol de administración personalizado reside en la cuenta de administrador, mientras que el rol de ejecución personalizado reside en cada cuenta de destino en la que desea crear StackSets utilizando los recursos definidos. A continuación, se activan usuarios y grupos específicos para que utilicen el rol de administración personalizado al llevar a cabo operaciones con StackSets.

Por ejemplo, puede crear los roles de administración personalizados A, B y C en la cuenta de administrador. Los usuarios y grupos con permiso para usar el rol A podrán crear StackSets que contienen los recursos de pila enumerados expresamente en el rol de ejecución personalizado X, pero no los enumerados en los roles Y o Z ni tampoco ningún recurso que no se incluya en ningún rol de ejecución.



Al actualizar un StackSet, el usuario debe especificar explícitamente un rol de administración personalizado, aunque sea el mismo que se utilizó con este StackSet anteriormente. CloudFormation lleva a cabo la actualización mediante el rol de administración personalizado especificado, siempre y cuando el usuario tenga permisos para llevar a cabo operaciones en ese StackSet.

Del mismo modo, el usuario también puede especificar una función de ejecución personalizada. Si especifica un rol de ejecución personalizado, CloudFormation utiliza dicho rol para actualizar

la pila, siempre que se cumplan los requisitos anteriores. Si el usuario no especifica un rol de ejecución personalizado, CloudFormation lleva a cabo la actualización mediante el rol de ejecución personalizado asociado anteriormente al StackSet, siempre y cuando el usuario tenga permisos para llevar a cabo operaciones en ese StackSet.

Administrator account

Cree un rol de administración personalizado en su cuenta de administrador, tal y como se detalla en [Control de los usuarios que pueden llevar a cabo operaciones con StackSets en cuentas de destino específicas](#). Incluya una relación de confianza entre el rol de administración personalizado y los roles de ejecución personalizados que deba utilizar.

Example Ejemplo de política de permisos

El siguiente ejemplo es una política de permisos para `AWSCloudFormationStackSetExecutionRole` que se define para la cuenta de destino, además de un rol de ejecución personalizado.

```
{
  "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

En las cuentas de destino en las que desee crear sus StackSets, cree un rol de ejecución personalizado que conceda permisos a aquellos servicios y recursos que los usuarios y grupos deban poder incluir en los StackSets.

Example Ejemplo de política de permisos

En el siguiente ejemplo, se proporcionan los permisos mínimos para los StackSets, junto con el permiso para crear tablas de Amazon DynamoDB.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudformation:*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "dynamodb:createTable"
      ],
      "Resource": "*"
    }
  ]
}
```

Example Política de confianza de ejemplo

Debe proporcionar la siguiente política de confianza a la hora de crear el rol para definir la relación de confianza.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/customized_admin_role"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

```
]
}
```

Configuración de permisos para operaciones con StackSets concretas

Además, puede configurar los permisos relativos a qué usuarios y grupos pueden llevar a cabo operaciones con StackSets concretas; por ejemplo, crear, actualizar o eliminar StackSets o instancias de las pilas. Para obtener más información, consulte [Acciones, recursos y claves de condición de CloudFormation](#) en la Referencia de autorizaciones de servicio.

Configuración de claves globales para mitigar problemas de adjuntos confusos

El problema de la sustitución confusa es un problema de seguridad en el que una entidad que no tiene permiso para realizar una acción puede obligar a una entidad con más privilegios a realizar la acción. En AWS, la suplantación entre servicios puede dar lugar al problema de la sustitución confusa. La suplantación entre servicios puede producirse cuando un servicio (el servicio que lleva a cabo las llamadas) llama a otro servicio (el servicio al que se llama). El servicio que lleva a cabo las llamadas se puede manipular para utilizar sus permisos a fin de actuar en función de los recursos de otro cliente de una manera en la que no debe tener permiso para acceder. Para evitarlo, AWS proporciona herramientas que le ayudan a proteger sus datos para todos los servicios con entidades principales de servicio a las que se les ha dado acceso a los recursos de su cuenta.

Se recomienda utilizar las claves de contexto de condición global [aws:SourceArn](#) y [aws:SourceAccount](#) en las políticas de recursos para limitar los permisos que CloudFormation StackSets concede a otro servicio para el recurso. Si se utilizan ambas claves contextuales de condición global, el valor `aws:SourceAccount` y la cuenta del valor `aws:SourceArn` deben utilizar el mismo ID de cuenta cuando se utilicen en la misma declaración de política.

La forma más eficaz de protegerse contra el problema de la sustitución confusa es utilizar la clave de contexto de condición global de `aws:SourceArn` con el ARN completo del recurso. Si no conoce el ARN completo del recurso o si especifica varios recursos, utiliza la clave de condición de contexto global `aws:SourceArn` con comodines (*) para las partes desconocidas del ARN. Por ejemplo, `arn:aws:cloudformation::123456789012:*`. Siempre que sea posible, use `aws:SourceArn`, que es más específico. Use `aws:SourceAccount` solo cuando no se pueda determinar el patrón del ARN o el ARN correcto.

Cuando StackSets asume el rol Administration (Administración) en su cuenta de administrador, StackSets rellena su ID de cuenta de administrador y el nombre de recurso de Amazon (ARN) de StackSets. Por lo tanto, puede definir las condiciones de [claves globales](#) `aws:SourceAccount` y

`aws:SourceArn` en las relaciones de confianza para evitar [problemas del suplente confuso](#). En el siguiente ejemplo, se muestra cómo se pueden utilizar las claves contextuales de condición global `aws:SourceArn` y `aws:SourceAccount` para evitar el problema del adjunto confundido.

Administrator account

Example Claves globales de `aws:SourceAccount` y `aws:SourceArn`

Al usar `StackSets`, defina las claves globales `aws:SourceAccount` y `aws:SourceArn` en su política de confianza `AWSCloudFormationStackSetAdministrationRole` para evitar problemas de suplente confuso.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "cloudformation.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "111122223333"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:cloudformation:*:111122223333:stackset/*"
        }
      }
    }
  ]
}
```

Example ARN de `StackSets`

Especifica sus ARN de `StackSets` asociados para un control más preciso.

JSON

```
{
```

```

"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Principal": {
      "Service": "cloudformation.amazonaws.com"
    },
    "Action": "sts:AssumeRole",
    "Condition": {
      "StringEquals": {
        "aws:SourceAccount": "111122223333",
        "aws:SourceArn": [
          "arn:aws:cloudformation:STACKSETS-
REGION:111122223333:stackset/STACK-SET-ID-1",
          "arn:aws:cloudformation:STACKSETS-
REGION:111122223333:stackset/STACK-SET-ID-2"
        ]
      }
    }
  }
]
}

```

Activación del acceso de confianza para StackSets con AWS Organizations

En este tema se proporcionan instrucciones sobre cómo activar el acceso de confianza con AWS Organizations, que los StackSets requieren para implementarse en todas las cuentas y Regiones de AWS mediante los permisos administrados por servicios. Para usar permisos autoadministrados, consulte en su lugar [Concesión de permisos autoadministrados](#).

Antes de crear un StackSet con permisos administrados por servicios, primero debe completar las siguientes tareas:

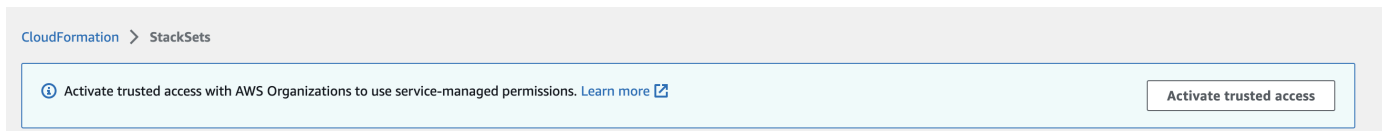
- [Habilite todas las características](#) en AWS Organizations. Solo con las características de facturación unificada habilitadas, no puede crear un StackSet con permisos administrados por servicios.
- Active el acceso de confianza con AWS Organizations. Esta acción permite que CloudFormation cree un rol vinculado al servicio en la cuenta de administración. Una vez activado el acceso de confianza, cuando crea un StackSet con permisos administrados por el servicio, CloudFormation crea tanto el rol vinculado al servicio necesario como un rol de servicio llamado `stacksets-exec-*` en las cuentas de destino (miembro).

Con el acceso de confianza activado, la cuenta de administración y las cuentas de administrador delegado pueden crear y administrar StackSets administrados por servicios para su organización.

Para activar el acceso de confianza, debe ser un usuario administrador en la cuenta de administración. Un usuario administrador es un usuario con permisos completos de su Cuenta de AWS. Para obtener más información, consulte [Crear un usuario administrador](#) en la Guía de referencia de AWS Account Management. Para obtener recomendaciones a fin de proteger la seguridad de la cuenta de administración, consulte [Prácticas recomendadas para la cuenta de administración](#) en la Guía del usuario de AWS Organizations.

Para activar el acceso de confianza

1. Inicie sesión en AWS como administrador de la cuenta de administración y abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation>.
2. En el panel de navegación, seleccione StackSets (Conjuntos de pilas). Si el acceso de confianza está desactivado, aparece un banner que le pide que active el acceso de confianza.



3. Seleccione Activar el acceso de confianza.

El acceso de confianza se activa correctamente cuando aparece el siguiente banner.



Note

Activar el acceso a organizaciones es lo mismo que habilitar el acceso a organizaciones y desactivar el acceso a organizaciones es lo mismo que deshabilitar el acceso a organizaciones. Estos términos se actualizaron en función de las directrices de marketing.

Para desactivar el acceso de confianza

Consulte [Conceptos de StackSets de CloudFormation y AWS Organizations](#) en la Guía del usuario de AWS Organizations.

Antes de desactivar el acceso de confianza con AWS Organizations, debe anular el registro de todos los administradores delegados. Para obtener más información, consulte [Registrar un administrador delegado](#).

 Note

Para obtener información sobre el uso de operaciones de API en lugar de la consola para activar o desactivar el acceso de confianza, consulte:

- [ActivateOrganizationsAccess](#)
- [DeactivateOrganizationsAccess](#)
- [DescribeOrganizationsAccess](#)

Roles vinculados a servicios

La cuenta de administración usa el rol vinculado al servicio `AWSServiceRoleForCloudFormationStackSetsOrgAdmin`. Puede modificar o eliminar este rol solo si el acceso de confianza con AWS Organizations está desactivado.

Cada cuenta de destino usa un rol vinculado al servicio `AWSServiceRoleForCloudFormationStackSetsOrgMember`. Puede modificar o eliminar este rol solo bajo dos condiciones: si el acceso de confianza con AWS Organizations está desactivado o si la cuenta se ha eliminado de la organización o unidad organizativa (OU) de destino.

Para obtener más información, consulte [CloudFormation StackSets y AWS Organizations](#) en la Guía del usuario de AWS Organizations.

Registro de una cuenta miembro como administrador delegado

Además de la cuenta de administración de su organización, las cuentas de miembro con permisos de administrador delegado pueden crear y administrar StackSets con permisos administrados por servicios para la organización. Los StackSets con permisos administrados por servicios se crean en la cuenta de administración, incluidos los StackSets creados por administradores delegados. Para registrarse como administrador delegado de la organización, la cuenta de miembro debe estar en ella. Para obtener más información acerca de cómo unirse a una organización, consulte [Invitación de una Cuenta de AWS a unirse a la organización](#).

La organización puede tener hasta cinco administradores delegados registrados a la vez. Los administradores delegados pueden optar por implementar en todas las cuentas de la organización o en unidades de organización específicas. El acceso de confianza con AWS Organizations debe activarse antes de que los administradores delegados puedan implementar en cuentas administradas por Organizations. Para obtener más información, consulte [Activación del acceso de confianza para StackSets con AWS Organizations](#).

 Important

Tenga en cuenta lo siguiente:

- Los administradores delegados tienen permisos completos para implementar en cuentas de la organización. La cuenta de administración no puede limitar los permisos del administrador delegado para la implementación en UO específicas ni para llevar a cabo operaciones específicas con StackSets.
- Asegúrese de que los administradores delegados tengan permisos `organizations:ListDelegatedAdministrators` para evitar posibles errores.

Puede registrar administradores delegados para la organización en las siguientes regiones: Este de EE. UU. (Ohio), Este de EE. UU. (Norte de Virginia), Oeste de EE. UU. (Norte de California), Oeste de EE. UU. (Oregón), Asia-Pacífico (Bombay), Asia Pacífico (Seúl), Asia Pacífico (Singapur), Asia Pacífico (Sídney), Asia Pacífico (Tokio), Canadá (centro), Europa (Fráncfort), Europa (Irlanda), Europa (Londres), Europa (París), Europa (Estocolmo), Israel (Tel Aviv), América del Sur (São Paulo), AWS GovCloud (Este de EE. UU.) y AWS GovCloud (Oeste de EE. UU.).

Puede registrar y anular el registro de administradores delegados mediante la [CloudFormation consola](#), [AWS CLI](#) o [los Kits de desarrollo de software \(Software Development Kit, SDK\) de AWS](#).

Para agregar un administrador delegado (consola)

1. Inicie sesión en AWS como administrador de la cuenta de administración y abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation/>.
2. En el panel de navegación, seleccione StackSets (Conjuntos de pilas).
3. En Administradores delegados, elija Registrar administrador delegado.
4. En el cuadro de diálogo Registrar administrador delegado, elija Registrar administrador delegado.

El mensaje de operación satisfactoria indica que la cuenta de miembro se ha registrado correctamente como administrador delegado.

Para anular el registro de un administrador delegado (consola)

1. Inicie sesión en AWS como administrador de la cuenta de administración y abra la consola de CloudFormation en <https://console.aws.amazon.com/>.
2. En el panel de navegación, seleccione StackSets (Conjuntos de pilas).
3. En Administradores delegados, seleccione la cuenta que desea anular del registro y, a continuación, elija Anular registro.

El mensaje de operación satisfactoria indica que la cuenta de miembro se ha anulado correctamente como administrador delegado.

Puede volver a registrar esta cuenta en cualquier momento.

Para registrar un administrador delegado (AWS CLI)

1. Abra la AWS CLI.
2. Ejecute el comando `register-delegated-administrator`.

```
$ aws organizations register-delegated-administrator \
  --service-principal=member.org.stacksets.cloudformation.amazonaws.com \
  --account-id="memberAccountId"
```

3. Ejecute el `list-delegated-administrators` comando para comprobar que la cuenta de miembro especificada se registra correctamente como administrador delegado.

```
$ aws organizations list-delegated-administrators \
  --service-principal=member.org.stacksets.cloudformation.amazonaws.com
```

Para anular el registro de un administrador delegado (AWS CLI)

1. Abra la AWS CLI.
2. Ejecute el comando `deregister-delegated-administrator`.

```
$ aws organizations deregister-delegated-administrator \
  --service-principal=member.org.stacksets.cloudformation.amazonaws.com \
  --account-id="memberAccountId"
```

3. Ejecute el `list-delegated-administrators` comando para comprobar que la cuenta de miembro especificada se anula correctamente como administrador delegado.

```
$ aws organizations list-delegated-administrators \
  --service-principal=member.org.stacksets.cloudformation.amazonaws.com
```

Puede volver a registrar esta cuenta en cualquier momento.

Introducción a los StackSets con una plantilla de ejemplo

Este tutorial lo ayudará a familiarizarse con los StackSets con la Consola de administración de AWS. Se explica cómo crear un StackSet con una plantilla de ejemplo. Obtendrá información sobre cómo implementar pilas en varias regiones, supervisar las operaciones con StackSets y ver los resultados.

En este tutorial, creará un StackSet que habilite AWS Config en su Cuenta de AWS en la región Oeste de EE. UU. (Oregón) (us-west-2) y la región Este de EE. UU. (Norte de Virginia) (us-east-1). Con los StackSets, puede crear, actualizar o eliminar pilas en varias cuentas y regiones con una sola operación, lo que lo convierte en una solución ideal para administrar la infraestructura a escala. Aunque en este tutorial se usa una sola cuenta por motivos de simplicidad, se demuestra de manera efectiva las capacidades multirregionales de los StackSets.

La plantilla de ejemplo está disponible en el siguiente bucket de S3: <https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/EnableAWSConfig.yml>.

Note

Los StackSets son gratuitos, pero se le cobrarán los recursos de AWS que cree con ellos, como AWS Config en este tutorial. Para más información, consulte [Precios de AWS Config](#).

Temas

- [Requisitos previos](#)
- [Creación de un StackSet con una plantilla de ejemplo desde la consola](#)

- [Supervisión de la creación de un StackSet](#)
- [Visualización de los resultados del StackSet](#)
- [Actualización del StackSet](#)
- [Cómo agregar pilas al StackSet](#)
- [Limpieza](#)
- [Siguiendo pasos](#)

Requisitos previos

Antes de iniciar este tutorial, asegúrese de que cumple los siguientes requisitos previos:

- Debe haber configurado los roles de IAM necesarios para los permisos autoadministrados. Para crear un StackSet e implementar pilas en una sola cuenta, necesita los siguientes roles en su cuenta:
 - `AWSCloudFormationStackSetAdministrationRole`
 - `AWSCloudFormationStackSetExecutionRole`

Para obtener instrucciones detalladas sobre cómo configurar estos roles, consulte [Concesión de permisos autoadministrados](#).

Creación de un StackSet con una plantilla de ejemplo desde la consola

Creación de un StackSet que habilite AWS Config

1. Abra la [consola de CloudFormation](#).
2. En la barra de navegación de la parte superior de la pantalla, elija la Región de AWS desde la cual desea administrar el StackSet.

Puede elegir cualquier región que admita StackSets. La región que seleccione no afecta a las regiones en las que puede llevar a cabo implementaciones con su StackSet.

3. En el panel de navegación, seleccione StackSets (Conjuntos de pilas).
4. En la parte superior de la página Conjuntos de pilas, seleccione Crear conjunto de pilas.
5. En Permisos, elija Permisos de autoservicio y los roles de IAM que ha creado en los requisitos previos.

- Para el rol de administrador de IAM, elija `AWSCloudFormationStackSetAdministrationRole`.
 - Para el nombre del rol de ejecución de IAM, elija `AWSCloudFormationStackSetExecutionRole`.
6. En Prerequisite - Prepare template (Requisito previo: preparar plantilla), elija Use a sample template (Usar una plantilla de ejemplo).
 7. En Seleccionar una plantilla de muestra, elija la plantilla Habilitar AWS Config. A continuación, elija Siguiente.

Esta plantilla crea los recursos necesarios para habilitar AWS Config en su cuenta, incluidos un registro de configuración y un canal de entrega.

8. En la página Especificar detalles del StackSet, en Nombre del StackSet, ingrese **my-awsconfig-stackset**.
9. En Descripción del StackSet, ingrese **A StackSet that enables Config across multiple Regions**.
10. En Parámetros, defina la configuración de AWS Config de la siguiente manera:
 - a. En Admitir todo tipo de recursos, mantenga el valor predeterminado, correcto, para registrar todos los tipos de recursos admitidos.
 - b. En Incluir tipos de recursos globales, mantenga el valor predeterminado, incorrecto, para excluir los recursos globales, como los roles de IAM.
 - c. Deje Lista de tipos de recursos si no son todos compatibles establecida en <Todos>.
 - d. En La región que contiene el recurso de rol vinculado al servicio Config, sustituya <DeployToAnyRegion> por **us-west-2**.

Esto significa que el rol vinculado al servicio denominado `AWSServiceRoleForConfig` solo se creará si se implementa una pila en la región Oeste de EE. UU (Oregón). Elegirá las regiones de implementación más adelante en este procedimiento.

- e. En Configuration recorder recording frequency, elija el registro DAILY.
11. Elija Siguiente para continuar.
 12. En la página Configurar las opciones del StackSet, elija Agregar etiqueta nueva y especifique un par de clave y valor para agregar una etiqueta:
 - a. En Clave, escriba **Stage**.
 - b. En Valor, introduzca **Test**.

Las etiquetas que se aplican a los StackSets se aplican a los recursos creados por las pilas.

13. En Configuración de ejecución, elija Activo para habilitar la gestión optimizada de las operaciones de CloudFormation:

- Las operaciones sin conflictos se ejecutan simultáneamente para acelerar los tiempos de implementación.
- Las operaciones en conflicto se ponen en cola y se procesan automáticamente en el orden en que se solicitaron.

Mientras las operaciones se estén ejecutando o estén en cola, CloudFormation pone en cola todas las operaciones entrantes (incluso si no están en conflicto). No puede cambiar la configuración de ejecución durante este tiempo.

14. Elija Siguiente.

15. En la página Establecer opciones de implementación, en Agregar pilas al StackSet, elija Implementar pilas nuevas.

16. Para Accounts (Cuentas), seleccione Deploy stacks in accounts (Implementar pilas en cuentas).

17. En el cuadro de texto, ingrese el ID de su Cuenta de AWS.

18. En Especificar regiones, seleccione las siguientes regiones en este orden:

1. Región Oeste de EE. UU. (Oregón) (us-west-2)
2. Región Este de EE. UU. (Norte de Virginia) (us-east-1)

Use la flecha hacia arriba que aparece junto a la región Oeste de EE. UU. (Oregón) para moverla a la primera posición de la lista si es necesario. El orden de las regiones determina su orden de implementación.

19. En Opciones de implementación, defina la siguiente configuración:

- a. En Maximum concurrent accounts (Máximo de cuentas simultáneas), mantenga los valores predeterminados de Number (Número) y 1.

En el caso de las implementaciones en varias cuentas, esta configuración significa que CloudFormation implementa la pila en solo una cuenta a la vez.

- b. Para Tolerancia a errores, mantenga los valores predeterminados de Número y 0.

Esto significa que puede fallar un máximo de cero implementaciones de la pila en una de las regiones especificadas antes de que CloudFormation detenga la implementación en la región actual y cancele la implementación en las regiones restantes.

- c. En Concurrencia de región, elija Secuencial (valor predeterminado).

Esta configuración garantiza que CloudFormation complete las implementaciones en una región antes de pasar a la siguiente.

- d. En Modo de simultaneidad, mantenga el valor predeterminado de Tolerancia a errores estricta.

En el caso de las implementaciones en varias cuentas, esto reduce el nivel de simultaneidad de cuentas cuando se producen errores, manteniéndose dentro de Tolerancia a errores +1.

20. Elija Siguiente.

21. En la página Review (Revisar), revise las opciones seleccionadas. Para realizar cambios, elija Editar en la sección correspondiente.

22. Cuando lo tenga todo listo para crear su StackSet, elija Enviar.

Supervisión de la creación de un StackSet

Al elegir Enviar, CloudFormation comienza a crear su StackSet y a implementar las pilas en las regiones especificadas de su cuenta. La página de detalles del StackSet se abre automáticamente, donde puede supervisar el progreso de la operación.

Supervisión de la creación del StackSet

1. En la página de detalles del StackSet, se muestra la pestaña Operaciones de forma predeterminada, en la que aparece la operación actual en curso.
2. El estado de la operación debe ser RUNNING inicialmente. CloudFormation crea pilas en las regiones que especificó de acuerdo con las opciones de implementación que configuró.
3. Para ver más detalles acerca de la operación, seleccione el ID de la operación en la lista.
4. En la página de detalles de la operación, puede ver el estado de las instancias de pila que se están creando en cada región.
5. Espere a que el estado de la operación cambie a SUCCEEDED, lo que indica que el StackSet y todas sus instancias de pila se crearon correctamente.

Visualización de los resultados del StackSet

Una vez completada la creación del StackSet, puede ver las instancias de pila implementadas y comprobar que AWS Config se ha habilitado en su cuenta en las regiones especificadas.

Visualización de los resultados del StackSet

1. En la página de detalles del StackSet, elija la pestaña Instancias de pila.
2. Debería ver una lista de las instancias de pila que se crearon en su cuenta en las regiones especificadas. Cada instancia de pila debe tener el estado SUCCEEDED, lo que indica que se implementó correctamente.
3. Para comprobar si AWS Config se habilitó en su cuenta, puede consultar la consola de AWS Config en cada una de las regiones implementadas.

Actualización del StackSet

Después de crear el StackSet, es posible que desee actualizarlo para modificar los valores de los parámetros o agregar más regiones. En esta sección se indica cómo actualizar el parámetro de frecuencia de registro de AWS Config.

Actualización del StackSet

1. En la página StackSets, seleccione su **my-awsconfig-stackset**.
2. Con el StackSet seleccionado, elija Editar detalles de StackSet en el menú Acciones.
3. En la página Elegir una plantilla, en Requisito previo: preparar la plantilla, elija Utilizar la plantilla actual.
4. Elija Siguiente.
5. En la página Especificar detalles del StackSet, en Parámetros, busque Configuration recorder recording frequency y cambie el valor de DAILY a CONTINUOUS.
6. Elija Siguiente.
7. En la página Configurar las opciones del StackSet, deje la configuración tal como está y elija Siguiente.
8. En la página Establecer opciones de implementación, especifique el ID de su cuenta y las mismas regiones que utilizó al crear el StackSet.
9. En Opciones de implementación, mantenga la misma configuración que antes.
10. Elija Siguiente.

11. En la página Revisar, revise los cambios y seleccione Enviar.
12. CloudFormation empieza a actualizar el StackSet. Puede supervisar el progreso en la pestaña Operaciones de la página de detalles del StackSet.

Cómo agregar pilas al StackSet

Puede agregar más pilas a su StackSet implementándolas en otras regiones. En esta sección se muestra cómo agregar pilas a una nueva región.

Cómo agregar pilas al StackSet

1. En la página StackSets, seleccione su **my-awsconfig-stackset**.
2. Con el StackSet seleccionado, elija Agregar pilas al StackSet en el menú Acciones.
3. En la página Establecer opciones de implementación, en Agregar pilas al StackSet, elija Implementar pilas nuevas.
4. En Cuentas, elija Implementar pilas en cuentas e ingrese el ID de su cuenta.
5. En Especificar regiones, seleccione una nueva región, como Europa (Irlanda) (eu-west-1).
6. En Opciones de implementación, mantenga la misma configuración que antes.
7. Elija Siguiente.
8. En la página Especificar anulaciones, deje los valores de las propiedades tal y como se especifican y elija Siguiente.
9. En la página Review (Revisar), revise las opciones y seleccione Submit (Enviar).
10. CloudFormation comienza a crear nuevas pilas en la región especificada. Puede supervisar el progreso en la pestaña Operaciones de la página de detalles del StackSet.

Limpieza

Para evitar que se generen cargos por recursos de AWS Config no deseados, debe llevar a cabo una limpieza; para ello, elimine las pilas de su StackSet, el propio StackSet y los roles de IAM que creó para este tutorial. Como todos los recursos están implementados en su cuenta, la limpieza es sencilla.

Eliminación de pilas del StackSet

1. En la página StackSets, seleccione su **my-awsconfig-stackset**.

2. Una vez seleccionado el StackSet, elija Eliminar pilas de StackSet en el menú Acciones.
3. En la página Establecer opciones de implementación, en Cuentas, elija Implementar pilas en cuentas e ingrese el ID de su cuenta.
4. En Especificar regiones, seleccione todas las regiones en las que ha implementado pilas.
5. En Opciones de implementación, mantenga la configuración predeterminada.
6. Asegúrese de que la opción Retener pilas no esté activada para que las pilas y sus recursos se eliminen.
7. Elija Siguiente.
8. En la página Review (Revisar), revise las opciones y seleccione Submit (Enviar).
9. CloudFormation comienza a eliminar las pilas de su StackSet. Puede supervisar el progreso en la pestaña Operaciones de la página de detalles del StackSet.

Eliminación del StackSet

1. Una vez eliminadas todas las pilas, en la página StackSets, seleccione su **my-awsconfig-stackset**.
2. Con el StackSet seleccionado, elija Eliminar StackSet en el menú Acciones.
3. Cuando se le pida confirmación, elija Eliminar.

Eliminación de los roles de servicio de IAM

Como solo llevó a cabo la implementación en su cuenta, solo tiene que eliminar los roles de IAM de esta única cuenta, lo que simplifica mucho más la limpieza en comparación con las implementaciones en varias cuentas.

1. Abra la [consola de IAM](#).
2. En el panel de navegación, elija Roles.
3. En el cuadro de búsqueda, escriba **AWSCloudFormationStackSet** para buscar los roles que creó para este tutorial.
4. Seleccione la casilla de verificación que hay junto a **AWSCloudFormationStackSetAdministrationRole**.
5. Elija Eliminar en la parte superior de la página.
6. En el cuadro de diálogo de confirmación, escriba **delete** y elija Eliminar.

7. Repita el mismo proceso para eliminar `AWSCloudFormationStackSetExecutionRole`.

Después de eliminar el StackSet, quedará un bucket de Amazon S3 en cada Región de AWS debido al atributo `DeletionPolicy` del recurso `AWS::S3::Bucket`. Esto preserva los datos del historial de AWS Config. Si ya no necesita estos datos, puede eliminar el bucket de forma manual y segura. Para eliminar un bucket, primero debe vaciarlo. Al vaciar un bucket, se eliminan todos los objetos que incluye.

Vaciado y eliminación de buckets de Amazon S3

1. Abra la [consola de Amazon S3](#).
2. En el panel de navegación del lado izquierdo de la consola, elija Buckets (Buckets).
3. En la lista Buckets, verá los buckets creados para este StackSet en cada región en la que llevó a cabo la implementación. Seleccione la opción situada junto al nombre del bucket creado para este StackSet y, a continuación, elija Empty.
4. En la página Vaciar bucket, confirme que desea vaciar el bucket escribiendo **permanently delete** en el campo de texto y, a continuación, elija Vaciar.
5. Supervise el progreso del proceso de vaciado del bucket en la página Vaciado del bucket: estado.
6. Para volver a la lista de buckets, seleccione Salir.
7. Seleccione la opción situada junto al nombre del bucket y, a continuación, seleccione Eliminar.
8. Cuando se le pida la confirmación, escriba el nombre del bucket y, a continuación, seleccione Eliminar bucket.
9. Supervise el progreso del proceso de eliminación del bucket en la lista Buckets. Cuando Amazon S3 completa la eliminación del bucket, lo elimina de la lista.
10. Repita este proceso para cada bucket creado por el StackSet en las diferentes regiones.

Siguientes pasos

¡Enhorabuena! Ha creado correctamente un StackSet con una plantilla de ejemplo, ha implementado pilas en varias regiones de su cuenta, ha actualizado el StackSet, ha agregado más pilas y ha limpiado sus recursos. Al centrarse en la implementación en una sola cuenta, ha simplificado el proceso de limpieza y, al mismo tiempo, ha aprendido las capacidades multirregionales principales de los StackSets.

Para obtener más información sobre los StackSets, consulte los siguientes temas:

- [Anulación de los valores de los parámetros en pilas en el StackSet de CloudFormation](#): obtenga información sobre cómo anular los valores de los parámetros para cuentas y regiones específicas.
- [Creación de StackSets de CloudFormation con permisos administrados por servicios](#): explore la creación de StackSets para implementaciones en varias cuentas con AWS Organizations.

Creación de StackSets de CloudFormation con permisos autoadministrados

En este tema se describe cómo crear StackSets con permisos autoadministrados para implementar pilas en todas las regiones y Cuentas de AWS.

Note

Antes de continuar, cree los roles de servicio de IAM que los StackSets necesitan para establecer una relación de confianza entre la cuenta desde la que está administrando el StackSet y la cuenta en la que está implementando pilas. Para obtener más información, consulte [Concesión de permisos autoadministrados](#).

Temas

- [Creación de un StackSet con permisos autoadministrados \(consola\)](#)
- [Creación de un StackSet con permisos autoadministrados \(AWS CLI\)](#)

Creación de un StackSet con permisos autoadministrados (consola)

Creación de un StackSet

1. Inicie sesión en la Consola de administración de AWS y abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation>.
2. En la barra de navegación de la parte superior de la pantalla, elija la Región de AWS desde la cual desea administrar el StackSet.
3. En el panel de navegación, seleccione StackSets (Conjuntos de pilas).
4. En la parte superior de la página Conjuntos de pilas, seleccione Crear conjunto de pilas.

5. En Permisos, elija Permisos de autoservicio y los roles de IAM que ha creado.
6. En Requisito previo - Preparación de la plantilla, seleccione Template is ready (La plantilla está lista).
7. En Especificar plantilla, elija especificar la dirección URL del bucket de S3 que contiene la plantilla de pila o cargar un archivo de plantilla de pila. A continuación, elija Siguiente.
8. En la página Especificar detalles del StackSet, proporcione un nombre para el StackSet, especifique los parámetros y, a continuación, elija Siguiente.
9. Elija Siguiente para continuar.
10. En la página Configurar opciones de StackSet, en Etiquetas, especifique las etiquetas que se aplicarán a los recursos de la pila. Para obtener más información acerca del uso de etiquetas en AWS, consulte [Organización y rastreo de costos mediante las etiquetas de asignación de costos de AWS](#) en la Guía del usuario de Administración de facturación y costos de AWS.
11. En Configuración de ejecución, elija Activo para habilitar la gestión optimizada de las operaciones de CloudFormation:
 - Las operaciones sin conflictos se ejecutan simultáneamente para acelerar los tiempos de implementación.
 - Las operaciones en conflicto se ponen en cola y se procesan automáticamente en el orden en que se solicitaron.

Mientras las operaciones se estén ejecutando o estén en cola, CloudFormation pone en cola todas las operaciones entrantes (incluso si no están en conflicto). No puede cambiar la configuración de ejecución durante este tiempo.

12. Si la plantilla contiene recursos de IAM, en Capacidades elija Confirmando que esta plantilla pueda crear recursos de IAM para especificar que desea utilizar recursos de IAM en la plantilla. Para obtener más información, consulte [Reconocimiento de recursos de IAM en plantillas de CloudFormation](#).
13. Elija Siguiente.
14. En la página Establecer opciones de implementación, en Agregar pilas al StackSet, elija Implementar pilas nuevas.
15. Para Accounts (Cuentas), seleccione Deploy stacks in accounts (Implementar pilas en cuentas). Pegue sus números de Cuenta de AWS de destino en el cuadro de texto, separando los distintos números con comas.

 Note

Puede incluir el ID de su cuenta de administrador si también desea implementar pilas en esa cuenta.

16. En Especificar regiones, elija las regiones de las que desea eliminar pilas.
17. Para Opciones de implementación, haga lo siguiente:
 - Para Número máximo de cuentas simultáneas, especifique cuántas cuentas se procesan simultáneamente.
 - En Tolerancia a errores, especifique el número máximo de errores de cuenta permitidos por región. La operación se detendrá y no continuará en otras regiones cuando se alcance este límite.
 - Para Concurrencia de regiones, elija cómo procesar las regiones: Secuencial (una región a la vez) o Paralelo (varias regiones simultáneamente).
 - Para el Modo de concurrencia, elija cómo se comporta la concurrencia durante la ejecución de la operación.
 - Tolerancia a errores estricta: reduce el nivel de simultaneidad de la cuenta cuando se producen errores, manteniéndose dentro de Tolerancia a errores +1.
 - Tolerancia flexible a errores: mantiene el nivel de concurrencia especificado (el valor de Número máximo de cuentas simultáneas) independientemente de los errores.
18. Elija Siguiente.
19. En la página Review (Revisar), revise las opciones seleccionadas. Para realizar cambios, elija Editar en la sección correspondiente.
20. Cuando lo tenga todo listo para crear su StackSet, elija Enviar.

CloudFormation inicia la creación del StackSet. Vea el progreso y el estado de la creación de las pilas del StackSet en la página de detalles del StackSet que se abre al elegir Enviar.

Creación de un StackSet con permisos autoadministrados (AWS CLI)

Siga los pasos de esta sección para utilizar la AWS CLI a fin de hacer lo siguiente:

- Crear el contenedor del StackSet.
- Implementar las instancias de pila.

Creación de un StackSet

1. Utilice el comando [create-stack-set](#) para crear un nuevo StackSet llamado *my-stackset*. En el siguiente ejemplo se utiliza una plantilla almacenada en un bucket de S3 y se incluye un parámetro que establece un *KeyPairName* con el valor *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. Una vez finalizado el comando `create-stack-set`, ejecute el comando [list-stack-sets](#) para ver que se ha creado el StackSet. Debería ver el nuevo StackSet en los resultados.

```
aws cloudformation list-stack-sets
```

3. Use el comando [create-stack-instances](#) para implementar pilas en su StackSet. En el siguiente ejemplo se implementan pilas en dos Cuentas de AWS (*account_ID_1* y *account_ID_2*) en dos regiones (*us-west-2* y *us-east-1*).

Defina el procesamiento simultáneo de cuentas y otras preferencias de implementación con la opción `--operation-preferences`. Este ejemplo utiliza una configuración basada en el recuento. Considere que `MaxConcurrentCount` no debe superar `FailureToleranceCount` + 1. Para ajustes basados en porcentajes, utilice `FailureTolerancePercentage` o `MaxConcurrentPercentage` en su lugar.

```
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
```

Para obtener más información, consulte [CreateStackInstances](#) en la Referencia de la API de AWS CloudFormation.

4. Use el comando [describe-stack-set-operation](#) para verificar que las pilas se hayan creado correctamente. Para la opción `--operation-id`, especifique el ID de operación que se devolvió como parte de la salida de `create-stack-instances`.

```
aws cloudformation describe-stack-set-operation \  
  --stack-set-name my-stackset \  
  --operation-id operation_ID
```

Creación de StackSets de CloudFormation con permisos administrados por servicios

Con los permisos administrados por servicios puede implementar pilas en cuentas administradas por AWS Organizations en regiones específicas. Con este modelo, no es necesario crear roles de IAM en cada cuenta de destino y Región de AWS. CloudFormation crea los roles de IAM en su nombre. Para obtener más información, consulte [Activación del acceso de confianza](#).

Temas

- [Consideraciones](#)
- [Creación de un StackSet con permisos administrados por servicios \(consola\)](#)
- [Creación de un StackSet con permisos administrados por servicios \(AWS CLI\)](#)

Consideraciones

Antes de crear un StackSet con permisos administrados por servicios, tenga en cuenta lo siguiente:

- Los StackSets con permisos administrados por servicios se pueden iniciar desde la cuenta de administración de su organización o desde cuentas de administrador delegado, pero todas las operaciones se realizan desde la cuenta de administración.
- CloudFormation no implementa pilas en la cuenta de administración, incluso si esa cuenta es parte de su organización o pertenece a una unidad organizativa (UO).
- El StackSet puede dirigirse a toda la organización (incluye todas las cuentas) o a UO especificadas. Cuando un StackSet se dirige a una UO principal, incluye automáticamente cualquier UO secundaria. Cuando el StackSet se dirige a UO específicas, todas las cuentas de esas UO se incluyen de forma predeterminada. Sin embargo, puede dirigirse a cuentas específicas mediante las opciones de filtros de cuentas.
- Varios StackSets pueden dirigirse a la misma organización o unidad organizativa.
- No puede dirigir el StackSet a cuentas de fuera de la organización.

- Los permisos asignados a la entidad principal de IAM (usuario, rol o grupo) que utiliza para iniciar sesión en la cuenta de administración determinan si está autorizado para la implementación con StackSets. Para ver un ejemplo de una política de IAM que concede permiso de implementación en una organización, consulte [Restringir las operaciones de conjuntos de pilas según la región y los tipos de recursos](#).
- Los administradores delegados tienen permisos completos para implementar en cuentas de la organización. La cuenta de administración no puede limitar los permisos del administrador delegado para la implementación en UO u operaciones StackSets específicas.
- La configuración de implementación automática se aplican en el nivel de StackSet. No puede ajustar las implementaciones automáticas selectivamente para unidades organizativas, cuentas o regiones.
- Los StackSets que utilizan permisos administrados por el servicio no admiten pilas anidadas ni plantillas que contengan macros o transformaciones.

Creación de un StackSet con permisos administrados por servicios (consola)

Creación de un StackSet

1. Inicie sesión en la Consola de administración de AWS y abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation>.
2. En la barra de navegación de la parte superior de la pantalla, elija la Región de AWS desde la cual desea administrar el StackSet.
3. En el panel de navegación, seleccione StackSets (Conjuntos de pilas).
4. En la parte superior de la página Conjuntos de pilas, seleccione Crear conjunto de pilas.
5. En Permisos, seleccione Permisos administrados por servicios.

Note

Si el acceso de confianza con AWS Organizations está deshabilitado, aparecerá un banner. Se requiere acceso de confianza para crear o actualizar un StackSet con permisos administrados por servicios. Solo el administrador de la cuenta de administración de la organización tiene permisos para [Activación del acceso de confianza para StackSets con AWS Organizations](#).

6. En Requisito previo - Preparación de la plantilla, seleccione Template is ready (La plantilla está lista).
7. En Especificar plantilla, elija especificar la dirección URL del bucket de S3 que contiene la plantilla de pila o cargar un archivo de plantilla de pila. A continuación, elija Siguiente.
8. En la página Especificar detalles del StackSet, proporcione un nombre para el StackSet, especifique los parámetros y, a continuación, elija Siguiente.
9. En la página Configurar opciones de StackSet, en Etiquetas, especifique las etiquetas que se aplicarán a los recursos de la pila. Para obtener más información acerca del uso de etiquetas en AWS, consulte [Organización y rastreo de costos mediante las etiquetas de asignación de costos de AWS](#) en la Guía del usuario de Administración de facturación y costos de AWS.
10. En Configuración de ejecución, elija Activo para habilitar la gestión optimizada de las operaciones de CloudFormation:
 - Las operaciones sin conflictos se ejecutan simultáneamente para acelerar los tiempos de implementación.
 - Las operaciones en conflicto se ponen en cola y se procesan automáticamente en el orden en que se solicitaron.

Mientras las operaciones se estén ejecutando o estén en cola, CloudFormation pone en cola todas las operaciones entrantes (incluso si no están en conflicto). No puede cambiar la configuración de ejecución durante este tiempo.

11. Si la plantilla contiene recursos de IAM, en Capacidades elija Confirmando que esta plantilla pueda crear recursos de IAM para especificar que desea utilizar recursos de IAM en la plantilla. Para obtener más información, consulte [Reconocimiento de recursos de IAM en plantillas de CloudFormation](#).
12. Seleccione Siguiente para continuar y activar el acceso de confianza si aún no está activado.
13. En la página Definir opciones de implementación, en Objetivos de implementación, realice una de las siguientes acciones:
 - Para realizar la implementación en todas las cuentas de la organización, elija Implementar en la organización.
 - Para realizar la implementación en todas las cuentas de unidades organizativas específicas, elija Implementar en unidades organizativas (OU). Elija Añadir una unidad organizativa, y, a continuación, pegue el ID de unidad organizativa de destino en el cuadro de texto. Repita para cada nueva unidad organizativa de destino.

Si selecciona Implementar en unidades organizativas (UO), para Tipo de filtro de cuentas, puede configurar sus objetivos de implementación para que sean cuentas individuales específicas al seleccionar una de las siguientes opciones y proporcionar los números de cuenta.

- Ninguno (predeterminado): implementa las pilas en todas las cuentas en las UO especificadas.
 - Intersección: implementa las pilas solo en cuentas individuales específicas dentro de las UO seleccionadas.
 - Diferencia: implementa las pilas en todas las cuentas de las UO seleccionadas, excepto en cuentas específicas.
 - Unión: implementa las pilas en las UO especificadas y en cuentas individuales adicionales.
14. En Implementación automática, elija si llevar a cabo la implementación automáticamente en cuentas que se agregan a la organización o UO de destino en el futuro. Para obtener más información, consulte [Habilitar o deshabilitar las implementaciones automáticas para StackSets en AWS Organizations](#).
15. Si ha habilitado la implementación automática, en Comportamiento de eliminación de cuentas, elija si los recursos de pila se conservan o eliminan cuando se quita una cuenta de una organización u unidad organizativa de destino.

 Note

Con la opción Retener pilas seleccionada, las pilas se eliminan del StackSet, pero las pilas y sus recursos asociados se retienen. Los recursos permanecen en su estado actual, pero ya no formarán parte del StackSet.

16. En Especificar regiones, elija las regiones de las que desea eliminar pilas.
17. Para Opciones de implementación, haga lo siguiente:
- Para Número máximo de cuentas simultáneas, especifique cuántas cuentas se procesan simultáneamente.
 - En Tolerancia a errores, especifique el número máximo de errores de cuenta permitidos por región. La operación se detendrá y no continuará en otras regiones cuando se alcance este límite.
 - Para Concurrencia de regiones, elija cómo procesar las regiones: Secuencial (una región a la vez) o Paralelo (varias regiones simultáneamente).

- Para el Modo de concurrencia, elija cómo se comporta la concurrencia durante la ejecución de la operación.
- Tolerancia a errores estricta: reduce el nivel de simultaneidad de la cuenta cuando se producen errores, manteniéndose dentro de Tolerancia a errores +1.
- Tolerancia flexible a errores: mantiene el nivel de concurrencia especificado (el valor de Número máximo de cuentas simultáneas) independientemente de los errores.
- Para las dependencias de StackSet, añada los ARN de StackSet dependientes, manteniéndose dentro de un máximo de 10 dependencias. Para obtener más información, consulte [Habilitar o deshabilitar las implementaciones automáticas para StackSets en AWS Organizations](#).

18. Elija Siguiente para continuar.

19. En la página Revisar, compruebe que el StackSet se implemente en las cuentas correctas de las regiones adecuadas y, a continuación, elija Crear StackSet.

Se abrirá la página Detalles de StackSet details. Puede ver el progreso y el estado de la creación de las pilas de su StackSet.

Creación de un StackSet con permisos administrados por servicios (AWS CLI)

Siga los pasos de esta sección para utilizar la AWS CLI a fin de hacer lo siguiente:

- Crear el contenedor del StackSet.
- Implementar las instancias de pila.

Note

Al actuar como administrador delegado, debe incluir `--call-as DELEGATED_ADMIN` en el comando.

Deploy to your organization

Creación de un StackSet

1. Utilice el comando [create-stack-set](#) para crear un nuevo StackSet llamado *my-stackset*. En el siguiente ejemplo se utiliza una plantilla almacenada en un bucket de S3, se habilitan las implementaciones automáticas y se conservan las pilas cuando se eliminan las cuentas. Para obtener más información, consulte [Habilitar o deshabilitar las implementaciones automáticas para StackSets en AWS Organizations](#).

```
aws cloudformation create-stack-set \  
  --stack-set-name my-stackset \  
  --template-url https://s3.region-code.amazonaws.com/amzn-s3-demo-bucket/MyApp.template \  
  --permission-model SERVICE_MANAGED \  
  --auto-deployment  
  Enabled=true,RetainStacksOnAccountRemoval=true,DependsOn=ARN1,ARN2
```

2. Use el comando [list-stack-sets](#) para confirmar que se creó el StackSet. El nuevo StackSet aparece en los resultados.

```
aws cloudformation list-stack-sets
```

- Si establece la opción `--call-as` a `DELEGATED_ADMIN` mientras ha iniciado sesión en su cuenta de miembro, `list-stack-sets` devuelve todos los StackSets con permisos administrados por servicios en la organización cuenta de administración.
 - Si establece la opción `--call-as` a `SELF` mientras mantiene una sesión iniciada en Cuenta de AWS, `list-stack-sets` devuelve todos los StackSets autoadministrados de su Cuenta de AWS.
 - Si establece la opción `--call-as` a `SELF` mientras ha iniciado sesión en la organización cuenta de administración, `list-stack-sets` devuelve todos los StackSets de la organización cuenta de administración.
3. Use el comando [create-stack-instances](#) para agregar pilas al StackSet. Para la opción `--deployment-targets`, especifique el ID raíz de la organización que se va a implementar en todas las cuentas de la organización.

Defina el procesamiento simultáneo de cuentas y otras preferencias de implementación con la opción `--operation-preferences`. Este ejemplo utiliza una configuración

basada en el recuento. Considere que `MaxConcurrentCount` no debe superar `FailureToleranceCount + 1`. Para ajustes basados en porcentajes, utilice `FailureTolerancePercentage` o `MaxConcurrentPercentage` en su lugar.

```
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
```

Para obtener más información, consulte [CreateStackInstances](#) en la Referencia de la API de AWS CloudFormation.

- Mediante el `operation-id` que se ha devuelto como parte de la salida de `create-stack-instances`, use el comando [describe-stack-set-operation](#) para comprobar que las pilas se han creado correctamente.

```
aws cloudformation describe-stack-set-operation \  
  --stack-set-name my-stackset \  
  --operation-id operation_ID
```

Deploy to organizational units (OUs)

Creación de un StackSet

- Utilice el comando [create-stack-set](#) para crear un nuevo StackSet llamado *my-stackset*. En el siguiente ejemplo se utiliza una plantilla almacenada en un bucket de S3 e incluye un parámetro que establece un *KeyPairName* con el valor *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
```

- Use el comando [list-stack-sets](#) para confirmar que se creó el StackSet. El nuevo StackSet aparece en los resultados.

```
aws cloudformation list-stack-sets
```

- Si establece la opción `--call-as` a `DELEGATED_ADMIN` mientras ha iniciado sesión en su cuenta de miembro, `list-stack-sets` devuelve todos los StackSets con permisos administrados por servicios en la organización cuenta de administración.
 - Si establece la opción `--call-as` a `SELF` mientras mantiene una sesión iniciada en Cuenta de AWS, `list-stack-sets` devuelve todos los StackSets autoadministrados de su Cuenta de AWS.
 - Si establece la opción `--call-as` a `SELF` mientras ha iniciado sesión en la organización cuenta de administración, `list-stack-sets` devuelve todos los StackSets de la organización cuenta de administración.
3. Use el comando [create-stack-instances](#) para agregar pilas al StackSet. Para la opción `--deployment-targets`, especifique los ID de las UO en los que se va a llevar a cabo la implementación.

Defina el procesamiento simultáneo de cuentas y otras preferencias de implementación con la opción `--operation-preferences`. Este ejemplo utiliza una configuración basada en el recuento. Considere que `MaxConcurrentCount` no debe superar `FailureToleranceCount + 1`. Para ajustes basados en porcentajes, utilice `FailureTolerancePercentage` o `MaxConcurrentPercentage` en su lugar.

```
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
```

Para obtener más información, consulte [CreateStackInstances](#) en la Referencia de la API de AWS CloudFormation.

4. Mediante el `operation-id` que se ha devuelto como parte de la salida de `create-stack-instances`, use el comando [describe-stack-set-operation](#) para comprobar que las pilas se han creado correctamente.

```
aws cloudformation describe-stack-set-operation \  
  --stack-set-name my-stackset \  
  --operation-id operation_ID
```

Deploy to specific accounts in OUs

Puede dirigirse a unidades organizativas (UO) específicas y utilizar el filtrado de cuentas para controlar con precisión qué cuentas reciben implementaciones de pila. De forma predeterminada, las pilas se implementan en todas las cuentas de las UO especificadas si no se especifica ningún filtrado de cuentas.

En la AWS CLI, especifique el filtrado de cuentas con la opción `--deployment-targets`. Para obtener más información, consulte [DeploymentTargets](#).

Después de crear el contenedor del StackSet con el comando `create-stack-set`, use uno de los siguientes ejemplos para implementar pilas en cuentas específicas.

Cómo dirigirse a cuentas específicas de una UO

En el siguiente ejemplo se implementan pilas solo en las cuentas A1 y A2 de 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
```

Exclusión de cuentas de una UO

En el siguiente ejemplo se implementan pilas en todas las cuentas de OU1, excepto en las cuentas A1 y 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
```

Habilitar o deshabilitar las implementaciones automáticas para StackSets en AWS Organizations

CloudFormation puede implementar automáticamente pilas adicionales en las nuevas cuentas de AWS Organizations cuando se agregan a la organización o unidades organizativas (UO) de destino. Puede habilitar las implementaciones automáticas y elegir si desea eliminar o conservar las pilas y

sus recursos asociados cuando se eliminen las cuentas de las UO de destino. Esta configuración se puede modificar en cualquier momento para los StackSets con permisos administrados por servicios.

Cómo funcionan las implementaciones automáticas

Cuando las implementaciones automáticas están habilitadas, se activarán cuando se agreguen las cuentas a una organización u unidad organizativa de destino, se eliminen de ellas o se muevan entre unidades organizativas de destino.

Por ejemplo, consideremos que StackSet1 apunta a OU1 en la región us-east-1, y que StackSet2 apunta a OU2 en la región us-east-1. OU1 contiene AccountA.

Si pasa AccountA de OU1 a OU2 con implementaciones automáticas habilitadas, CloudFormation ejecuta automáticamente una operación de eliminación para eliminar la pila StackSet1 de AccountA y pone en cola una operación de creación que agrega la pila StackSet2 a AccountA.

Consideraciones

A continuación, se detallan consideraciones que se deben tener en cuenta cuando se utilizan implementaciones automáticas:

- La característica de implementación automática se habilita en el nivel de StackSet. No puede ajustar las implementaciones automáticas selectivamente para unidades organizativas, cuentas o regiones.
- Los valores de parámetro sustituidos solo se aplican a las cuentas que están actualmente en las unidades organizativas de destino y a sus unidades organizativas secundarias. Las cuentas agregadas a las UO de destino y sus UO secundarias en el futuro utilizarán los valores predeterminados del StackSet y no los valores anulados.
- Las implementaciones automáticas no tienen en cuenta los filtros de segmentación a nivel de cuenta. Si se dirige a cuentas específicas y habilita implementaciones automáticas, su StackSet continuará las implementaciones en las cuentas añadidas recientemente en la organización implementada. Para evitar las implementaciones en las cuentas recién agregadas, desactive las implementaciones automáticas.
- Administración de dependencias: defina hasta 10 dependencias por StackSet, con hasta 100 dependencias por cuenta. Por ejemplo, si tiene cinco StackSets con cinco dependencias cada uno, tiene 25 dependencias que cuentan para el límite de 100 dependencias. Puede solicitar un aumento del límite a través de la [consola de Service Quotas](#). Las dependencias se eliminan cuando se eliminan los StackSets o se desactivan las organizaciones.

- Se recomienda habilitar la ejecución administrada cuando se utilizan implementaciones automáticas. La ejecución administrada permitirá que las operaciones de implementación automática en varias cuentas de destino se ejecuten simultáneamente dentro de un StackSet, lo que aumentará la velocidad de procesamiento, especialmente en el caso de las organizaciones más grandes.

Habilitar o deshabilitar implementaciones automáticas (consola)

Habilitación o deshabilitación de implementaciones automáticas

1. Inicie sesión en la Consola de administración de AWS y abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation>.
2. En la barra de navegación de la parte superior de la pantalla, elija la Región de AWS en la que creó el StackSet.
3. En el panel de navegación, seleccione StackSets (Conjuntos de pilas).
4. En la página StackSets, seleccione la opción situada junto al nombre del StackSet que desee actualizar.
5. Elija Editar implementación automática en el menú Acciones en la esquina superior derecha.
6. En el cuadro de diálogo que se abre, haga lo siguiente:
 - a. En Implementación automática, elija Activado o Desactivado.
 - b. En Comportamiento de eliminación de cuentas, seleccione Eliminar pilas o Retener pilas. Los recursos retenidos permanecen en su estado actual, pero ya no formarán parte del StackSet.
 - c. Para las dependencias de StackSet, añada los ARN de StackSet dependientes, manteniéndose dentro de un máximo de 10 dependencias.
7. Seleccione Save.

Habilitar o deshabilitar implementaciones automáticas (AWS CLI)

Habilitación o deshabilitación de implementaciones automáticas

1. Utilice el comando [update-stack-set](#) con la opción `--auto-deployment`.

El siguiente comando habilita las implementaciones automáticas.

```
aws cloudformation update-stack-set --stack-set-name my-stackset \  
  --use-previous-template --auto-deployment  
  Enabled=true,RetainStacksOnAccountRemoval=true,DependsOn=ARN1,ARN2
```

Como alternativa, para deshabilitar las implementaciones automáticas, especifique `Enabled=false` como el valor de la opción `--auto-deployment`, como en el siguiente ejemplo.

```
aws cloudformation update-stack-set --stack-set-name my-stackset \  
  --use-previous-template --auto-deployment Enabled=false
```

- Mediante el ID de la operación que se ha devuelto como parte de la salida de `update-stack-set`, ejecute [describe-stack-set-operation](#) para comprobar que el StackSet se haya actualizado correctamente.

```
aws cloudformation describe-stack-set-operation --operation-id operation_ID
```

Actualización de StackSets de CloudFormation

Puede actualizar el StackSet mediante la consola de CloudFormation o la AWS CLI.

Para agregar y eliminar cuentas y regiones de un StackSet, consulte [Adición de pilas a StackSets](#) y [Eliminar pilas de los StackSets](#). Para anular los valores de los parámetros para una pila, consulte [Anulación de parámetros en pilas](#).

Temas

- [Actualización del StackSet \(consola\)](#)
- [Actualización del StackSet \(AWS CLI\)](#)

Actualización del StackSet (consola)

Actualización de un StackSet

- Inicie sesión en la Consola de administración de AWS y abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation>.

2. En la barra de navegación de la parte superior de la pantalla, elija la Región de AWS en la que creó el StackSet.
 3. En el panel de navegación, seleccione StackSets (Conjuntos de pilas).
 4. En la página StackSet, seleccione el StackSet que desea actualizar.
 5. Con el StackSet seleccionado, elija Editar detalles de StackSet en el menú Acciones.
 6. En la página Elegir una plantilla, actualice la sección Permisos según sea necesario o pase al siguiente paso.
 7. En Requisito previo: preparar plantilla, elija Usar plantilla actual para usar la plantilla actual o Reemplazar plantilla actual para especificar una URL de S3 a otra plantilla o cargar una nueva plantilla.
 8. Elija Siguiente.
 9. En la página Especificar detalles del StackSet, en Descripción del StackSet, actualice la descripción del StackSet según sea necesario.
 10. En Parámetros, actualice los valores de los parámetros según sea necesario.
 11. Elija Siguiente.
 12. En la página Configurar opciones del conjunto de pilas, en Etiquetas, modifique etiquetas según sea necesario. Puede añadir, actualizar o eliminar etiquetas. Para obtener más información acerca del uso de etiquetas en AWS, consulte [Organización y rastreo de costos mediante las etiquetas de asignación de costos de AWS](#) en la Guía del usuario de Administración de facturación y costos de AWS.
 13. Para la configuración de ejecución, puede actualizar la configuración de ejecución según sea necesario.
- 
- Note**
- Recuerde que no puede cambiar la configuración de ejecución cuando las operaciones están en ejecución o en cola.
14. Si la plantilla contiene recursos de IAM, en Capacidades elija Confirmando que esta plantilla pueda crear recursos de IAM para especificar que desea utilizar recursos de IAM en la plantilla. Para obtener más información, consulte [Reconocimiento de recursos de IAM en plantillas de CloudFormation](#).
 15. Elija Siguiente.

16. En la página Establecer opciones de implementación, proporcione las cuentas y regiones para la actualización.

CloudFormation implementará actualizaciones de pilas en las cuentas especificadas en la primera región, a continuación pasará a la siguiente, y así sucesivamente, mientras los fallos de implementación de una región no superen la tolerancia de fallos especificada.

- a. [Permisos autoadministrados] En Cuentas, Ubicaciones de implementación, elija Implementar pilas en cuentas. Pegue los ID de las cuentas de destino que ha utilizado para crear el StackSet en el cuadro de texto, separando varios números mediante comas.

[Permisos administrados por el servicio] Realice una de las siguientes acciones:

- Elija Implementar en unidades organizativas (OU). Ingrese las UO de destino que haya utilizado para crear el StackSet.
 - Elija Implementar en cuentas. Pegue los ID de las UO o las cuentas de destino que haya utilizado para crear el StackSet.
- b. En Especificar regiones, especifique el orden en el que desea que CloudFormation implemente las actualizaciones.
- c. Para Opciones de implementación, haga lo siguiente:
- Para Número máximo de cuentas simultáneas, especifique cuántas cuentas se procesan simultáneamente.
 - En Tolerancia a errores, especifique el número máximo de errores de cuenta permitidos por región. La operación se detendrá y no continuará en otras regiones cuando se alcance este límite.
 - Para Concurrencia de regiones, elija cómo procesar las regiones: Secuencial (una región a la vez) o Paralelo (varias regiones simultáneamente).
 - Para el Modo de concurrencia, elija cómo se comporta la concurrencia durante la ejecución de la operación.
 - Tolerancia a errores estricta: reduce el nivel de simultaneidad de la cuenta cuando se producen errores, manteniéndose dentro de Tolerancia a errores +1.
 - Tolerancia flexible a errores: mantiene el nivel de concurrencia especificado (el valor de Número máximo de cuentas simultáneas) independientemente de los errores.
 - [Permisos administrados por el servicio] Para las dependencias de StackSet, añada los ARN de StackSet dependientes, manteniéndose dentro de un máximo de

10 dependencias. Para obtener más información, consulte [Habilitar o deshabilitar las implementaciones automáticas para StackSets en AWS Organizations](#).

- d. Elija Siguiente para continuar.
17. En la página Review (Revisar), revise las opciones seleccionadas. Para realizar cambios, elija Editar en la sección correspondiente.
18. Cuando esté listo para continuar, elija Enviar.

CloudFormation comienza a aplicar las actualizaciones en el StackSet y muestra la pestaña Operaciones de la página de detalles del StackSet. Puede ver el progreso y el estado de las operaciones de actualización en la pestaña Operations (Operaciones).

Actualización del StackSet (AWS CLI)

Note

Al actuar como administrador delegado, debe incluir `--call-as DELEGATED_ADMIN` en el comando.

1. Actualización de un StackSet

Ejecute el comando de [update-stack-set](#) para hacer cambios en el StackSet.

En los siguientes ejemplos, actualizamos el StackSet mediante la opción `--parameters`. Específicamente, modificamos la frecuencia de entrega de instantánea predeterminada para la configuración del canal de entrega de `TwentyFour_Hours` a `Twelve_Hours`. Dado que seguimos utilizando la plantilla actual, añadimos la opción `--use-previous-template`.

Defina el procesamiento simultáneo de cuentas y otras preferencias de implementación con la opción `--operation-preferences`. Estos ejemplos utilizan una configuración basada en el recuento. Considere que `MaxConcurrentCount` no debe superar `FailureToleranceCount` + 1. Para ajustes basados en porcentajes, utilice `FailureTolerancePercentage` o `MaxConcurrentPercentage` en su lugar.

[Permisos autoadministrados] Para la opción `--accounts`, proporcione los ID de cuenta a los que desea que se dirija la actualización.

```
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
```

[Permisos administrados por el servicio] Para la opción `--deployment-targets`, proporcione el ID de raíz de la organización o los ID de unidad organizativa (UO) a los que desea que se dirija la actualización.

```
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
```

Para obtener más información, consulte [UpdateStackSet](#) en la Referencia de la API de AWS CloudFormation.

- Compruebe que el StackSet se haya actualizado correctamente al ejecutar el comando `describe-stack-set-operation` para mostrar el estado y los resultados de la operación de actualización. Para `--operation-id`, utilice el ID de operación devuelto con el comando `update-stack-set`.

```
aws cloudformation describe-stack-set-operation \  
--operation-id operation_ID
```

Adición pilas a CloudFormation StackSets

Al crear un StackSet, puede crear las pilas de ese StackSet. CloudFormation también le permite agregar más pilas para otras cuentas y regiones en cualquier momento después de crear el StackSet. Puede añadir pilas con la consola de CloudFormation o con la AWS CLI.

Temas

- [Cómo agregar pilas a un StackSet \(consola\)](#)
- [Cómo agregar pilas a un StackSet \(AWS CLI\)](#)

Cómo agregar pilas a un StackSet (consola)

Cómo agregar pilas a un StackSet

1. Inicie sesión en la Consola de administración de AWS y abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation>.
2. En la barra de navegación de la parte superior de la pantalla, elija la Región de AWS en la que creó el StackSet.
3. En el panel de navegación, seleccione StackSets (Conjuntos de pilas). En la página StackSet, seleccione el StackSet que creó.
4. Con el StackSet seleccionado, elija Agregar pilas al StackSet en el menú Acciones.
5. En la página Establecer opciones de implementación, realice las siguientes acciones:
 - a. En Agregar pilas al StackSet, seleccione Implementar pilas nuevas.
 - b. A continuación, lleve a cabo las siguientes acciones en función de la configuración de los permisos del StackSet:
 - [Permisos autoadministrados] En Cuentas, Ubicaciones de implementación, elija Implementar pilas en cuentas. Pegue sus números de cuenta de destino en el cuadro de texto, separando los distintos números con comas..
 - [Permisos administrados por el servicio] Para los Objetivos de implementación, realice una de las siguientes acciones:
 - Elija Implementar en la organización para realizar la implementación en todas las cuentas de la organización.
 - Elija Implementar en unidades organizativas (OU) para implementarlas en todas las cuentas de unidades organizativas específicas. Elija Añadir otra unidad organizativa, y, a continuación, pegue el ID de unidad organizativa de destino en el cuadro de texto. Repita para cada nueva unidad organizativa de destino. CloudFormation también se dirige a cualquier UO secundaria de los destinos seleccionados.

Note

Si agrega una UO a la que ya se dirige el StackSet, CloudFormation crea nuevas pilas en cualquier cuenta de la UO que aún no tenga pilas del StackSet (por ejemplo, cuentas que se agregaron a la UO después de crear el StackSet y con implementaciones automáticas deshabilitadas).

- c. En Especificar regiones, especifique en qué cuentas de destino Regiones de AWS desea realizar el despliegue en las cuentas de destino que especificó en el paso anterior. De forma predeterminada, CloudFormation implementará pilas en las cuentas especificadas en la primera región, a continuación pasará a la siguiente, y así sucesivamente, mientras los fallos de implementación de una región no superen la tolerancia de fallos especificada.
 - d. Para Opciones de implementación, haga lo siguiente:
 - Para Número máximo de cuentas simultáneas, especifique cuántas cuentas se procesan simultáneamente.
 - En Tolerancia a errores, especifique el número máximo de errores de cuenta permitidos por región. La operación se detendrá y no continuará en otras regiones cuando se alcance este límite.
 - Para Concurrencia de regiones, elija cómo procesar las regiones: Secuencial (una región a la vez) o Paralelo (varias regiones simultáneamente).
 - Para el Modo de concurrencia, elija cómo se comporta la concurrencia durante la ejecución de la operación.
 - Tolerancia a errores estricta: reduce el nivel de simultaneidad de la cuenta cuando se producen errores, manteniéndose dentro de Tolerancia a errores +1.
 - Tolerancia flexible a errores: mantiene el nivel de concurrencia especificado (el valor de Número máximo de cuentas simultáneas) independientemente de los errores.
 - e. Elija Siguiente.
6. En la página Specify Overrides (Especificar anulaciones), deje los valores de la propiedad tal y como aparecen. No anulará ningún valor de propiedad de las pilas que va a crear. Elija Siguiente.
 7. En la página Review (Revisar), revise las opciones seleccionadas. Para realizar cambios, elija Editar en la sección correspondiente.
 8. Cuando esté listo para continuar, elija Enviar.

CloudFormation inicia la creación de las pilas. Vea el progreso y el estado de la creación de las pilas del StackSet en la página de detalles del StackSet que se abre al elegir Enviar. Cuando haya finalizado, sus nuevas pilas deberían aparecer en la pestaña Stack instances (Instancias de pila).

Cómo agregar pilas a un StackSet (AWS CLI)

Note

Al actuar como administrador delegado, debe incluir `--call-as DELEGATED_ADMIN` en el comando.

Cómo agregar pilas a un StackSet con permisos autoadministrados

Utilice el comando CLI `create-stack-instances`. Para la opción `--accounts`, proporcione los ID de cuentas para los que desee crear pilas.

```
aws cloudformation create-stack-instances --stack-set-name my-stackset \  
--accounts account_id --regions eu-west-1 us-west-2
```

Cómo agregar pilas a un StackSet con permisos administrados por servicios

Utilice el comando CLI `create-stack-instances`. Para la opción `--deployment-targets`, proporcione el ID de organización (raíz) o los ID de UO para los que desee crear pilas. Para ver ejemplos de comandos que se dirigen a cuentas específicas, consulte [Creación de un StackSet con permisos administrados por servicios \(AWS CLI\)](#).

```
aws cloudformation create-stack-instances --stack-set-name my-stackset \  
--deployment-targets OrganizationalUnitIds=ou-rcuk-r1qi0wl7 --regions eu-west-1 us-west-2
```

Note

Si agrega una UO a la que ya se dirige el StackSet, CloudFormation crea nuevas pilas en cualquier cuenta de la UO que aún no tenga pilas del StackSet (por ejemplo, cuentas que se agregaron a la UO después de crear el StackSet y con implementaciones automáticas deshabilitadas).

Anulación de los valores de los parámetros en pilas en el StackSet de CloudFormation

En algunos casos, es posible que desee que pilas en determinadas regiones o cuentas tengan diferentes valores de propiedad que los especificados en el propio StackSet. Por ejemplo, es posible que desee especificar un valor diferente para un parámetro determinado en función de si una cuenta se utiliza para el desarrollo o la producción. En estas situaciones, CloudFormation permite anular los valores de los parámetros en pilas por cuenta y región. Puede anular los valores de los parámetros de plantilla al crear las pilas y puede anular los valores de parámetros de las pilas existentes. Solo puede establecer los parámetros que anuló previamente en las pilas con los valores originales especificados en el StackSet.

Las anulaciones de los valores de parámetros se aplican a las pilas de las cuentas y regiones que seleccione. Durante las actualizaciones del StackSet, no se actualiza ningún valor de parámetro anulado para una pila, sino que se retiene su valor anulado.

Solo puede anular valores de parámetros que se han especificado en el StackSet; para agregar o eliminar un parámetro, debe actualizar la plantilla del StackSet. Si agrega un parámetro a una plantilla de StackSet, para poder anular el valor de ese parámetro en una pila, primero debe actualizar todas las pilas con el nuevo parámetro y valor especificados en el StackSet. Una vez que todas las pilas se han actualizado con el nuevo parámetro, puede anular el valor del parámetro en pilas individuales como desee.

Para obtener información sobre cómo anular valores de parámetros de StackSets al crear pilas, consulte [Adición de pilas a StackSets](#).

Temas

- [Anulación de parámetros en pilas \(consola\)](#)
- [Anulación de parámetros en pilas \(AWS CLI\)](#)

Anulación de parámetros en pilas (consola)

Anulación de parámetros de pilas específicas

1. Inicie sesión en la Consola de administración de AWS y abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation>.

2. En la barra de navegación de la parte superior de la pantalla, elija la Región de AWS en la que creó el StackSet.
3. En el panel de navegación, seleccione StackSets (Conjuntos de pilas). En la página StackSets, seleccione el StackSet.
4. Con el StackSet seleccionado, elija Anulación de parámetros de StackSet en el menú Acciones.
5. En la página Establecer opciones de implementación, proporcione las cuentas y las regiones para las pilas para las que creará anulaciones.

De forma predeterminada, CloudFormation implementará pilas en las cuentas especificadas en la primera región, luego pasará a la siguiente, y así sucesivamente, mientras las fallas de implementación de una región no superen la tolerancia de fallos especificada.

- a. [Permisos autoadministrados] En Destinos de implementación, elija Implementar pilas en cuentas. Pegue algunos o todos los ID de cuenta de destino que utilizó para crear el StackSet.

[Permisos administrados por el servicio] Realice una de las siguientes acciones:

- Elija Implementar en unidades organizativas (OU). Ingrese una o más de las OU de destino que utilizó para crear el StackSet. Los valores de parámetro sustituidos solo se aplican a las cuentas que están actualmente en las unidades organizativas de destino y a sus unidades organizativas secundarias. Las cuentas agregadas a las OU de destino y sus OU secundarias en el futuro utilizarán los valores predeterminados del StackSet y no los valores anulados.
 - Elija Implementar en cuentas. Pegue algunos o todos los ID de OU o de cuenta de destino que utilizó para crear el StackSet.
- b. En Especificar regiones, agregue una o varias de las regiones en las que implementó las pilas para este StackSet.

Si añade varias regiones, el orden de las regiones bajo Especificar regiones determina su orden de implementación.

- c. Para Opciones de implementación, haga lo siguiente:
 - Para Número máximo de cuentas simultáneas, especifique cuántas cuentas se procesan simultáneamente.

- En Tolerancia a errores, especifique el número máximo de errores de cuenta permitidos por región. La operación se detendrá y no continuará en otras regiones cuando se alcance este límite.
- Para Concurrencia de regiones, elija cómo procesar las regiones: Secuencial (una región a la vez) o Paralelo (varias regiones simultáneamente).
- Para el Modo de concurrencia, elija cómo se comporta la concurrencia durante la ejecución de la operación.
 - Tolerancia a errores estricta: reduce el nivel de simultaneidad de la cuenta cuando se producen errores, manteniéndose dentro de Tolerancia a errores +1.
 - Tolerancia flexible a errores: mantiene el nivel de concurrencia especificado (el valor de Número máximo de cuentas simultáneas) independientemente de los errores.

d. Elija Siguiente.

6. En la página Especificar anulaciones, seleccione las casillas para los parámetros que desea anular, y luego elija Anular valor del conjunto de pilas en el menú Editar valor de anulación.
7. En la página Anular los valores de los parámetros del conjunto de pilas, realice los cambios y, a continuación, seleccione Guardar cambios.

 Note

Para reestablecer los parámetros anulados al valor original especificado en el StackSet, compruebe todos los parámetros y elija Establecer en el valor del StackSet del menú Editar valor de anulación. De esta forma, se eliminan todos los valores anulados cuando se actualizan las pilas.

8. En la página Review (Revisar), revise las opciones seleccionadas. Para realizar cambios, elija Editar en la sección correspondiente.
9. Cuando esté listo para continuar, elija Enviar.

CloudFormation empieza a actualizar las pilas. Vea el progreso y el estado de las pilas en la página de detalles del StackSet que se abre al elegir Enviar.

Anulación de parámetros en pilas (AWS CLI)

Note

Al actuar como administrador delegado, debe incluir `--call-as DELEGATED_ADMIN` en el comando.

Anulación de parámetros de pilas específicas

1. Utilice el comando [update-stack-instances](#) de la AWS CLI y especifique la opción `--parameter-overrides`.

[Permisos autoadministrados] Para la opción `--accounts`, proporcione los ID de cuentas para los que desee anular valores de parámetros en pilas.

```
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
```

[Permisos administrados por servicios] Para la opción `--deployment-targets`, proporcione el ID raíz de organización, los ID de unidad organizativa o los ID de cuenta de AWS Organizations para los que desee anular los parámetros de las pilas. En este ejemplo, anulamos los valores de parámetros para las pilas en todas las cuentas de la UO con el ID *ou-rcuk-1x5j1lwo*.

Los valores de parámetro reemplazados solo se aplican a las cuentas que están actualmente en la unidad organizativa de destino y a sus unidades organizativas secundarias. Las cuentas agregadas a la UO de destino y sus UO secundarias en el futuro utilizarán los valores predeterminados del StackSet y no los valores reemplazados.

```
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. Compruebe que los valores de los parámetros se reemplazaron correctamente en las pilas ejecutando el comando `describe-stack-set-operation` para mostrar el estado y los resultados de

la operación de actualización. Para `--operation-id`, utilice el ID de operación devuelto con el comando `update-stack-instances`.

```
aws cloudformation describe-stack-set-operation --operation-id operation_ID
```

Eliminación de pilas de CloudFormation StackSets

Puede eliminar las pilas de un StackSets mediante la consola de CloudFormation o la AWS CLI.

Temas

- [Eliminación de pilas del StackSet \(consola\)](#)
- [Eliminación de pilas del StackSet \(AWS CLI\)](#)

Note

Al eliminar pilas de una unidad organizativa (UO) de nivel superior, se elimina esa UO como objetivo del StackSet.

Eliminación de pilas del StackSet (consola)

Para eliminar pilas

1. Inicie sesión en la Consola de administración de AWS y abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation>.
2. En la barra de navegación de la parte superior de la pantalla, elija la Región de AWS en la que creó el StackSet.
3. En el panel de navegación, seleccione StackSets (Conjuntos de pilas). En la página StackSets, seleccione el StackSet.
4. Una vez seleccionado su StackSet, elija Eliminar pilas de StackSet en el menú Acciones.
5. En la página Establecer opciones de implementación, seleccione primero las cuentas y regiones en las que desea eliminar las pilas.
 - a. [Permisos autoadministrados] En Cuentas, seleccione Implementar pilas en cuentas, o Implementar pilas en unidades organizaciones.

Si elige Implementar pilas en cuentas, pegue sus números de cuenta de destino en el cuadro de texto Números de cuenta, separe los números con coma.

Si elige Implementar pilas en unidades organizativas, pegue un ID de UO de destino en el cuadro de texto Números de organización para seleccionar todas las cuentas que forman parte de la organización especificada.

- b. [Permisos administrados por el servicio] En el caso de las unidades organizativas (OU), especifique los ID de las OU de destino.

 Important

CloudFormation eliminará las pilas tanto de las unidades organizativas de destino especificadas como de sus unidades organizativas secundarias.

Para Tipo de filtro de cuentas, para definir qué cuentas tendrán pilas eliminadas elija una de las siguientes opciones y proporcione números de cuenta.

- Ninguno (predeterminado): elimina las pilas de todas las cuentas en las UO especificadas.
- Intersección: elimina las pilas solo de cuentas individuales específicas dentro de las UO seleccionadas.
- Diferencia: elimina las pilas de todas las cuentas de las UO seleccionadas, excepto de cuentas específicas.
- Unión: elimina las pilas de las UO especificadas y de las cuentas individuales adicionales.

- c. En Especificar regiones, elija las regiones de las cuentas objetivo de las que desea eliminar pilas.

6. Para Opciones de implementación, haga lo siguiente:

- Para Número máximo de cuentas simultáneas, especifique cuántas cuentas se procesan simultáneamente.
- En Tolerancia a errores, especifique el número máximo de errores de cuenta permitidos por región. La operación se detendrá y no continuará en otras regiones cuando se alcance este límite.

- En Retener pilas, habilite esta opción para guardar las pilas y sus recursos asociados al eliminarlos de su StackSet. Los recursos permanecerán en su estado actual, pero ya no formarán parte del StackSet.
 - Para Concurrencia de regiones, elija cómo procesar las regiones: Secuencial (una región a la vez) o Paralelo (varias regiones simultáneamente).
 - Para el Modo de concurrencia, elija cómo se comporta la concurrencia durante la ejecución de la operación.
 - Tolerancia a errores estricta: reduce el nivel de simultaneidad de la cuenta cuando se producen errores, manteniéndose dentro de Tolerancia a errores +1.
 - Tolerancia flexible a errores: mantiene el nivel de concurrencia especificado (el valor de Número máximo de cuentas simultáneas) independientemente de los errores.
7. Elija Siguiente.
 8. En la página Review (Revisar), revise las opciones seleccionadas. Para realizar cambios, elija Editar en la sección correspondiente.
 9. Cuando lo tenga todo listo para eliminar las pilas del StackSet, elija Enviar.

Después de eliminar las pilas, puede comprobar si se han eliminado las pilas del StackSet en la página de detalles del StackSet, en la pestaña Instancias de pila.

Eliminación de pilas del StackSet (AWS CLI)

Note

Al actuar como administrador delegado, debe incluir `--call-as DELEGATED_ADMIN` en el comando.

Use el comando `delete-stack-instances` con el nombre de su StackSet.

En estos ejemplos, usamos la opción `--no-retain-stacks`, porque no retenemos ninguna pila. Utilice `--retain-stacks` en vez de `--no-retain-stacks` si quiere conservar las pilas y sus recursos.

Para `--regions`, especifique las Regiones de AWS de las que desea eliminar pilas, por ejemplo, `us-west-2` y `us-east-1`.

Defina el procesamiento simultáneo de cuentas y otras preferencias con la opción `--operation-preferences`. Estos ejemplos utilizan una configuración basada en el recuento. Considere que `MaxConcurrentCount` no debe superar `FailureToleranceCount + 1`. Para ajustes basados en porcentajes, utilice `FailureTolerancePercentage` o `MaxConcurrentPercentage` en su lugar.

Eliminación de pilas (permisos autoadministrados)

Para la opción `--accounts`, especifique los ID de las cuentas de las que desee eliminar pilas.

```
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
```

Eliminación de pilas (permisos administrados por servicios)

Para `--deployment-targets`, especifique el ID de la organización raíz o los ID de unidad organizativa (UO) de los que desee eliminar pilas.

Important

CloudFormation eliminará las pilas tanto de las unidades organizativas de destino especificadas como de sus unidades organizativas secundarias.

```
aws cloudformation delete-stack-instances --stack-set-name my-stackset \  
  --deployment-targets OrganizationalUnitIds=ou-rcuk-1x5jlwo,ou-rcuk-slrl5lh0a \  
  --regions us-west-2 us-east-1 \  
  --no-retain-stacks \  
  --operation-preferences MaxConcurrentCount=1,FailureToleranceCount=0
```

Para más información, consulte [DeleteStackInstances](#) en la Referencia de la API AWS CloudFormation.

Opcionalmente, una vez que haya terminado la eliminación de pilas, compruebe que las pilas se hayan eliminado del StackSet ejecutando el comando `describe-stack-set-operation` para mostrar el estado y los resultados de la operación de eliminación de pilas. Para `--operation-id`, utilice el ID de operación devuelto con el comando `delete-stack-instances`.

```
aws cloudformation describe-stack-set-operation --stack-set-name my-stackset \  
--operation-id ddf16f54-ad62-4d9b-b0ab-3ed8e9example
```

Eliminación de StackSets de CloudFormation

Para eliminar un StackSet, primero debe eliminar todas las pilas del StackSet. Para obtener información sobre cómo eliminar todas las pilas, consulte [Eliminar pilas de los StackSets](#).

Temas

- [Eliminación de un StackSet \(consola\)](#)
- [Eliminación de un StackSet \(AWS CLI\)](#)
- [Eliminar roles de servicio \(opcional\)](#)

Eliminación de un StackSet (consola)

Eliminación de un StackSet

1. Inicie sesión en la Consola de administración de AWS y abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation>.
2. En la barra de navegación de la parte superior de la pantalla, elija la Región de AWS en la que creó el StackSet.
3. En la página StackSets, seleccione el StackSet.
4. Con el StackSet seleccionado, elija Eliminar StackSet en el menú Acciones.
5. Cuando se le pida que confirme que desea eliminar el StackSet, seleccione Eliminar StackSet.

Eliminación de un StackSet (AWS CLI)

Note

Al actuar como administrador delegado, debe incluir `--call-as DELEGATED_ADMIN` en el comando.

Eliminación de un StackSet

1. Use el siguiente comando `delete-stack-set`. Cuando se le solicite confirmación, escriba **y** y, a continuación, pulse Intro.

```
aws cloudformation delete-stack-set --stack-set-name my-stackset
```

2. Para comprobar que el StackSet se haya eliminado, ejecute el comando `list-stack-sets`. Los resultados del comando `list-stack-sets` deberían mostrar la pila con un estado de DELETED.

```
aws cloudformation list-stack-sets
```

Eliminar roles de servicio (opcional)

Si ya no necesita los roles de servicio de IAM que CloudFormation requiere para llevar a cabo operaciones con StackSets, le recomendamos que los elimine.

En el caso de los StackSets autogestionados, los roles que creó. Para obtener más información sobre estos roles, consulte [Concesión de permisos autoadministrados](#).

En el caso de los StackSets administrados por servicios, los roles que se crearon automáticamente para StackSets tiene el sufijo `CloudFormationStackSetsOrgAdmin` en la cuenta de administración de la organización y `CloudFormationStackSetsOrgMember` en cada cuenta de destino. Para obtener más información, consulte [Roles vinculados a servicios](#).

Eliminación de un rol de servicio (consola)

1. Inicie sesión en Consola de administración de AWS y abra la consola IAM en <https://console.aws.amazon.com/iam/>.
2. En el panel de navegación, elija Roles y, a continuación, marque la casilla de verificación junto al rol que desea eliminar.
3. En el menú Role actions (Acciones de rol) en la parte superior de la página, elija Delete role (Eliminar rol).
4. En el cuadro de diálogo de confirmación, elija Yes, Delete (Sí, eliminar). Si está seguro, puede continuar con la eliminación, aunque los datos del último acceso al servicio sigan cargándose.

Eliminación de un rol de servicio (AWS CLI)

- Use el siguiente comando `delete-role`. Cuando se le pida confirmación, escriba **y** y, a continuación, pulse Enter.

```
aws iam delete-role --role-name role name
```

Para obtener más información sobre la eliminación de roles, consulte [Eliminación de roles o perfiles de instancia](#) en la Guía de usuario de IAM.

Cómo evitar las implementaciones fallidas de StackSets mediante puertas de cuentas de destino

Una puerta de cuenta es una característica opcional que lo ayuda a verificar que una cuenta de destino cumpla determinados requisitos antes de que CloudFormation inicie las operaciones con StackSets en dicha cuenta. Esta verificación se lleva a cabo a través de una función de AWS Lambda que actúa como una comprobación previa.

Un ejemplo común de una puerta de cuenta es verificar que no haya alarmas de CloudWatch activas o sin resolver en la cuenta de destino. CloudFormation invoca la función de Lambda cada vez que inicie operaciones de pila en la cuenta de destino y solo continúa si la función devuelve el código `SUCCEEDED`. Si la función de Lambda devuelve el estado `FAILED`, CloudFormation no continúa con la operación solicitada. Si no tiene configurada una función de Lambda de puertas de cuentas, CloudFormation omite la comprobación y continúa con su operación.

Si su cuenta de destino falla una comprobación de puerta de cuenta, el número de operaciones fallidas se incrementa en el número o porcentaje de tolerancia de fallos de pilas. Para obtener más información acerca de las tolerancias ante fallos, consulte [Opciones de operaciones con StackSets](#).

La regulación de cuenta solo está disponible para operaciones de StackSets. Esta característica no está disponible para otras operaciones de CloudFormation más allá de StackSets.

Requisitos

Para las puertas de cuentas, se deben cumplir los siguientes requisitos:

- Su función de Lambda debe llamarse `AWSCloudFormationStackSetAccountGate` para utilizar esta característica.

- `AWSCloudFormationStackSetExecutionRole` debe tener permisos para invocar su función de Lambda. Sin estos permisos, CloudFormation omitirá la comprobación de puertas de cuentas y continuará con las operaciones de la pila.
- El permiso `InvokeFunction` de Lambda debe añadirse a las cuentas de destino para que funcione la regulación de la cuenta. La política de confianza de la cuenta de destino no tiene una relación de confianza con la cuenta del administrador. El siguiente es un ejemplo de declaración de política que otorga permisos `InvokeFunction` de Lambda.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "lambda:InvokeFunction"
      ],
      "Resource": "*"
    }
  ]
}
```

Plantillas de CloudFormation para crear funciones de Lambda

Utilice las siguientes plantillas de ejemplo para crear funciones

`AWSCloudFormationStackSetAccountGate` de Lambda. Para crear una nueva pila con una de estas plantillas, consulte [Creación de una pila en la consola de CloudFormation](#).

Ubicación de plantilla	Descripción
https://s3.amazonaws.com/cloudformation-sta-ckset-sample-templates-us-east-1/AccountGateSucceeded.yml	Crea una pila que implementa una función de puerta de cuenta de Lambda que devolverá un estado de <code>SUCCEEDED</code> .
https://s3.amazonaws.com/cloudformation-sta-ckset-sample-templates-us-east-1/AccountGateFailed.yml	Crea una pila que implementa una función de puerta de cuenta de Lambda que devolverá un estado de <code>FAILED</code> .

Elección del modo de simultaneidad para CloudFormation StackSets

El modo de simultaneidad es un parámetro de [StackSetOperationPreferences](#) que permite elegir cómo se comporta el nivel de simultaneidad durante las operaciones con StackSets. Puede elegir entre los siguientes modos:

- **Tolerancia estricta a errores:** esta opción reduce de forma dinámica el nivel de simultaneidad para garantizar que el número de cuentas fallidas nunca supere el valor de Tolerancia a errores + 1. La simultaneidad real inicial se establece en el valor más bajo entre el valor de Número máximo de cuentas simultáneas o el valor de Tolerancia a errores + 1. A continuación, la simultaneidad real se reduce proporcionalmente en función del número de errores. Este es el comportamiento predeterminado.
- **Tolerancia ligera a errores:** esta opción desacopla Tolerancia a errores de la simultaneidad real. Esto permite que las operaciones con StackSets se ejecuten en el nivel de simultaneidad establecido por el valor de Número máximo de cuentas simultáneas, independientemente del número de errores.

Tolerancia a errores estricta reduce la velocidad de implementación a medida que se producen errores en las operaciones con StackSets, ya que la simultaneidad disminuye en cada error.

Tolerancia a errores suave prioriza la velocidad de implementación y, al mismo tiempo, aprovecha las capacidades de seguridad de CloudFormation. Esto le permite revisar y abordar los errores de las operaciones con StackSets para detectar problemas comunes, como los relacionados con los recursos existentes, las cuotas de servicio y los permisos.

Para obtener más información sobre los errores en las operaciones de pila de StackSets, consulte [Motivos comunes para el fallo en una operación de la pila](#).

Para obtener más información sobre Número máximo de cuentas simultáneas y Tolerancia a errores, consulte [Opciones de operaciones con StackSets](#).

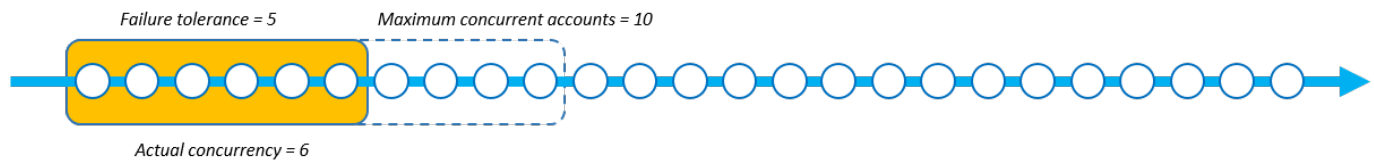
Cómo funciona cada modo de simultaneidad

Las siguientes imágenes proporcionan una representación visual de cómo funciona cada modo de simultaneidad durante una operación con StackSets. La cadena de nodos representa una implementación en una Región de AWS y cada nodo es una Cuenta de AWS objetivo.

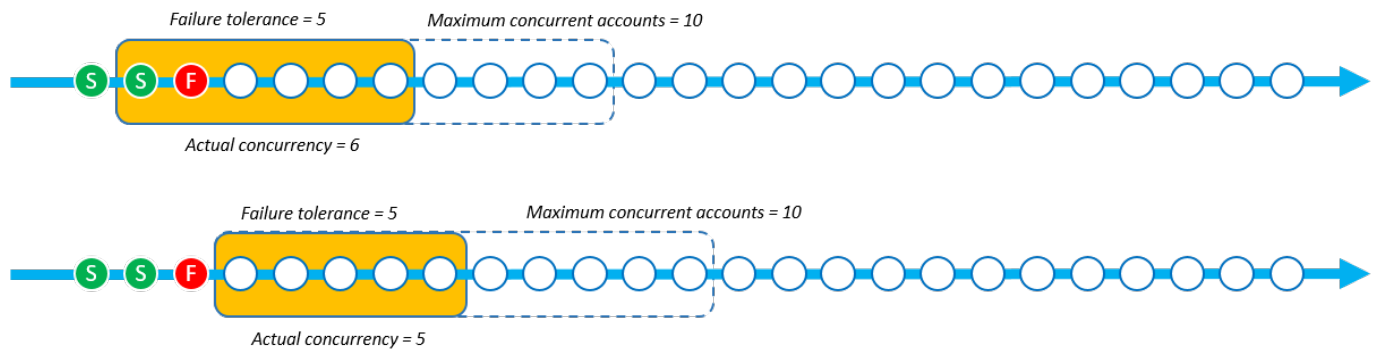
Tolerancia estricta a errores

Cuando una operación con StackSets que utiliza Tolerancia a errores estricta tiene el valor de Tolerancia a errores establecido en 5 y el valor de Número máximo de cuentas simultáneas en 10, la simultaneidad real es 6. La simultaneidad real es 6 porque el valor de Tolerancia a errores de 5 + 1 es inferior al valor de Número máximo de cuentas simultáneas.

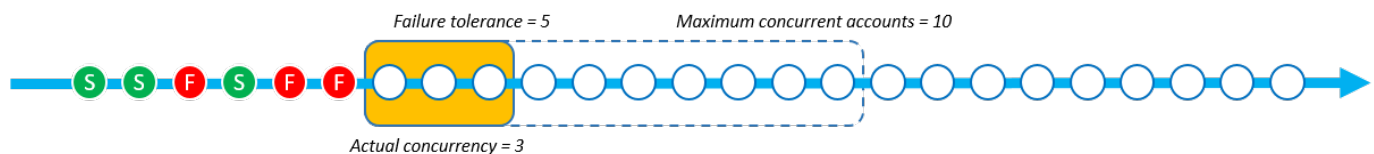
En la siguiente imagen se muestra el impacto que el valor de Tolerancia a errores tiene en el valor de Número máximo de cuentas simultáneas y el impacto que ambos tienen en la simultaneidad real de la operación con StackSets:



Cuando comienza la implementación y hay instancias de pila fallidas, la simultaneidad real se reduce para ofrecer una experiencia de implementación segura. La simultaneidad real se reduce de 6 a 5 cuando StackSets no logra implementar una instancia de pila.

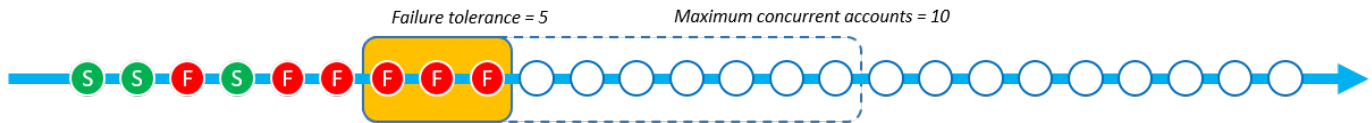


El modo Tolerancia estricta a errores reduce la simultaneidad real proporcionalmente al número de instancias de pila fallidas. En el siguiente ejemplo, la simultaneidad real se reduce de 5 a 3 cuando StackSets no logra implementar 2 instancias de pila más, lo que eleva el total de instancias de pila con errores a 3.



StackSets no puede completar la operación con StackSets cuando el número de instancias de pila con errores es igual al valor definido de Tolerancia a errores + 1. En el siguiente ejemplo,

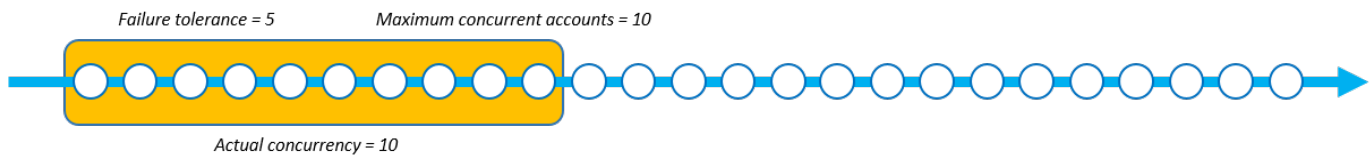
StackSets no realiza la operación cuando hay 6 instancias de pila con errores y el valor de Tolerancia a errores es 5.



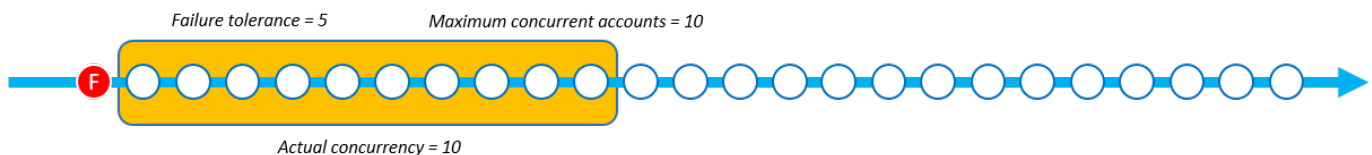
En este ejemplo, CloudFormation implementó 9 instancias de pila (3 correctas y 6 con errores) antes de detener la operación con StackSets.

Tolerancia ligera a errores

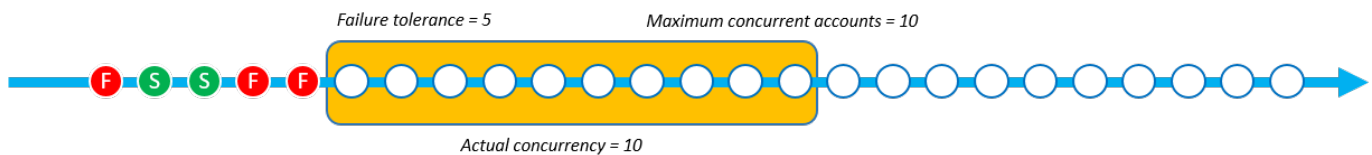
Cuando una operación con StackSets que utiliza Tolerancia a errores suave tiene el valor de Tolerancia a errores establecido en 5 y el valor de Número máximo de cuentas simultáneas en 10, la simultaneidad real es 10.



Cuando comienza la implementación y hay instancias de pila con errores la simultaneidad real no cambia. En el siguiente ejemplo, se produjo un error en una operación de pila, pero la simultaneidad real se mantiene en 10.

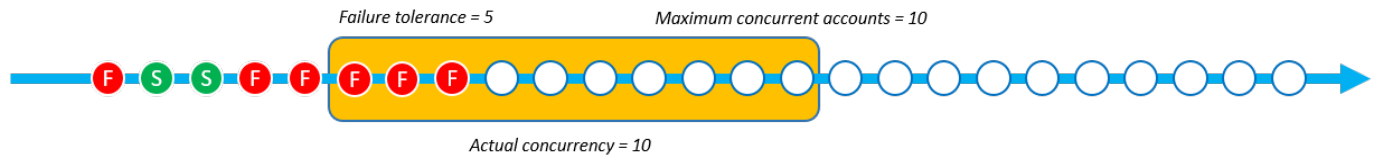


La simultaneidad real se mantiene en 10 incluso después de que se produzcan 2 errores más en las instancias de pila.

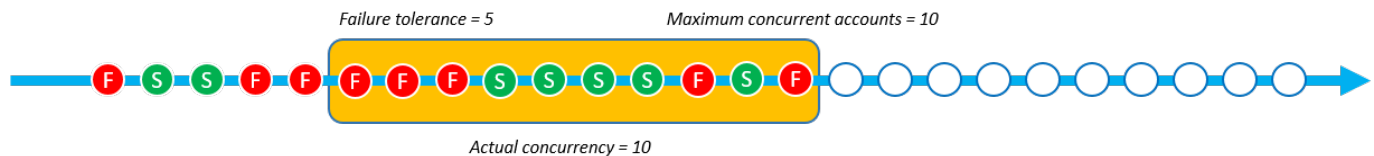


StackSets no puede completar la operación con StackSets cuando el número de instancias de pila con errores supera el valor de Tolerancia a errores. En el siguiente ejemplo, StackSets no realiza la operación cuando hay 6 instancias de pila con errores y el recuento de Tolerancia a

errores es 5. Sin embargo, la operación no finalizará hasta que se completen las operaciones restantes de la cola de simultaneidad.



StackSets sigue implementando instancias de pila que ya están en la cola de simultaneidad. Esto significa que la cantidad de instancias de pila con errores puede ser superior al valor de Tolerancia a errores. En el siguiente ejemplo, hay 8 instancias de pila con errores porque a la cola de simultaneidad aún le quedaban 7 operaciones por completar, a pesar de que la operación con StackSets había alcanzado el valor de Tolerancia a errores de 5.



En este ejemplo, StackSets implementó 15 instancias de pila (7 correctas y 8 con errores) antes de detener la operación de pila.

Elección entre Tolerancia estricta a errores y Tolerancia ligera a errores en función de la velocidad de implementación

La elección entre los modos Tolerancia a errores estricta y Tolerancia a errores suave depende de la velocidad preferida de implementación del StackSet y del número permitido de errores de implementación.

En las siguientes tablas, se muestra cómo gestiona cada modo de simultaneidad una operación con StackSets con errores al intentar implementar un total de 1000 instancias de pila. En cada escenario, el valor de Tolerancia a errores se establece en 100 instancias de pila y el valor de Número máximo de cuentas simultáneas se establece en 250 instancias de pila.

Si bien StackSets en realidad pone en cola las cuentas como una ventana deslizante (consulte [Cómo funciona cada modo de simultaneidad](#)), este ejemplo muestra la operación en lotes para demostrar la velocidad de cada modo.

Tolerancia estricta a errores

En este ejemplo, en el que se utiliza el modo Tolerancia estricta a errores, se reduce la simultaneidad real en relación con el número de errores que se producen en cada lote anterior. Cada lote tiene 20 instancias con errores, lo que reduce en 20 la simultaneidad real del lote siguiente hasta que la operación con StackSets alcance el valor de Tolerancia a errores de 100.

En la siguiente tabla, la simultaneidad real inicial del primer lote es de 101 instancias de pila. La simultaneidad real es 101 porque es el valor más bajo de Número máximo de cuentas simultáneas (250) o Tolerancia a errores (100) + 1. Cada lote contiene 20 implementaciones de instancias de pila con errores, lo que reduce la simultaneidad real de cada lote siguiente en 20 instancias de pila.

Tolerancia estricta a errores	Lote 1	Lote 2	Lote 3	Lote 4	Lote 5	Lote 6
Recuento de simultaneidad real	101	81	61	41	21	-
Recuento de instancias con errores	20	20	20	20	20	-
Recuento de instancias de pila correctas	81	61	41	21	1	-

La operación que utilizó Tolerancia a errores estricta completó 305 implementaciones de instancias de pila en 5 lotes cuando la operación con StackSets alcanzó el valor de Tolerancia a errores de 100 instancias de pila. La operación con StackSets implementa correctamente 205 instancias de pila antes de que se produzca un error.

Tolerancia ligera a errores

En este ejemplo, en el que se utiliza el modo Tolerancia ligera a errores, se mantiene el mismo recuento de simultaneidad real definido por el valor de Número máximo de cuentas simultáneas de 250 instancias de pila, independientemente del número de instancias con errores. Las operaciones con StackSets mantienen la misma simultaneidad real hasta que alcanzan el valor de Tolerancia a errores de 100 instancias.

En la siguiente tabla, la simultaneidad real inicial del primer lote es de 250 instancias de pila. La simultaneidad real es 250 porque el valor de Número máximo de cuentas simultáneas está establecido en 250 y el modo Tolerancia ligera a errores permite a StackSets utilizar este valor como la simultaneidad real, independientemente del número de errores. A pesar de que hay 50 errores en cada uno de los lotes de este ejemplo, la simultaneidad real no se ve afectada.

Tolerancia ligera a errores	Lote 1	Lote 2	Lote 3	Lote 4	Lote 5	Lote 6
Recuento de simultaneidad real	250	250	-	-	-	-
Recuento de instancias con errores	50	50	-	-	-	-
Recuento de instancias de pila correctas	200	200	-	-	-	-

Con el mismo valor de Número máximo de cuentas simultáneas y de Tolerancia a errores, la operación que utilizó el modo Tolerancia ligera a errores completó 500 implementaciones de

instancias de pila en 2 lotes. La operación con StackSets implementa correctamente 400 instancias de pila antes de que se produzca un error.

Elegir el modo de simultaneidad (consola)

Al crear o actualizar un StackSet, en la página Establecer opciones de implementación, en Modo de simultaneidad, seleccione Tolerancia a errores estricta o Tolerancia a errores suave.

Elegir el modo de simultaneidad (AWS CLI)

Puede utilizar el parámetro `ConcurrencyMode` con los siguientes comandos de StackSets:

- [create-stack-instances](#)
- [delete-stack-instances](#)
- [detect-stack-set-drift](#)
- [import-stacks-to-stack-set](#)
- [update-stack-instances](#)
- [update-stack-set](#)

Estos comandos tienen un parámetro existente llamado `--operation-preferences` que puede usar la configuración `ConcurrencyMode`. `ConcurrencyMode` se puede establecer en uno de los siguientes valores:

- `STRICT_FAILURE_TOLERANCE`
- `SOFT_FAILURE_TOLERANCE`

En el siguiente ejemplo, se crea una instancia de pila utilizando el `STRICT_FAILURE_TOLERANCE` `ConcurrencyMode`, con el valor de `FailureToleranceCount` establecido en 10 y el valor de `MaxConcurrentCount` establecido en 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

Si desea conocer los procedimientos detallados para crear y actualizar un StackSet, consulte los temas siguientes:

- [Creación de StackSets \(permisos autoadministrados\)](#)
- [Creación de StackSets \(permisos administrados por servicios\)](#)
- [Actualización de StackSets](#)

Detección de desviaciones en CloudFormation StackSets

Incluso aunque administre las pilas y los recursos que estas contienen a través de CloudFormation, los usuarios pueden cambiar esos recursos fuera de CloudFormation. Los usuarios pueden editar los recursos directamente usando el servicio subyacente que ha creado el recurso. Al llevar a cabo la detección de desviaciones en un StackSet, puede determinar si alguna de las instancias de pila pertenecientes a ese StackSet difiere, o se ha desviado, de su configuración esperada.

Temas

- [Cómo lleva a cabo CloudFormation la detección de desviaciones en un StackSet](#)
- [Detección de la desviación en un StackSet \(consola\)](#)
- [Detección de la desviación en un StackSet \(AWS CLI\)](#)
- [Detención de la detección de desviaciones en un StackSet](#)

Cómo lleva a cabo CloudFormation la detección de desviaciones en un StackSet

Cuando CloudFormation lleva a cabo la detección de desviaciones en un StackSet, la lleva a cabo en la pila asociada a cada instancia de pila del StackSet. Para ello, CloudFormation compara el estado actual de cada recurso de la pila con el estado esperado de dicho recurso, tal como se define en la plantilla de la pila y en los parámetros de entrada especificados. Si el estado actual de un recurso varía con respecto a su estado esperado, se considera que ese recurso se ha desviado. Si uno o más recursos en una pila se han desviado, entonces se considera que la pila en sí misma se ha desviado, y se considera que las instancias de pila con las que está asociada la pila también se han desviado. Si una o más instancias de pila de un StackSet se han desviado, se considera que el StackSet se ha desviado.

La detección de desviaciones identifica los cambios no administrados; es decir, los cambios realizados en las pilas fuera de CloudFormation. Los cambios efectuados directamente en una pila a través de CloudFormation, en lugar de en el nivel del StackSet, no se consideran una desviación. Por ejemplo, supongamos que tiene una pila asociada a una instancia de pila de un StackSet. Si usa CloudFormation para actualizar esa pila para usar una plantilla diferente, eso no se considera desviación, aunque esa pila ahora tiene una plantilla diferente a cualquier otra pila perteneciente al StackSet. Esto se debe a que la pila todavía coincide con la plantilla y la configuración de parámetros esperados en CloudFormation.

Para obtener información detallada sobre cómo CloudFormation realiza la detección de desviaciones en una pila, consulte [Detectar cambios de configuración no administrados en pilas y recursos con detección de derivación](#).

Dado que CloudFormation realiza la detección de desviaciones en cada pila individualmente, tiene en cuenta cualquier valor de parámetro anulado al determinar si una pila se ha desviado. Para obtener más información sobre la anulación de parámetros de plantilla en instancias de pila, consulte [Anulación de los valores de los parámetros en pilas en el StackSet de CloudFormation](#).

Si realiza la detección de desviaciones [directamente en una pila](#) asociada a una instancia de pila, esos resultados de desviaciones no estarán disponibles en la página de consola de StackSets.

Detección de la desviación en un StackSet (consola)

Detección de la desviación en un StackSet

1. Abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation>.
2. En la página StackSets, seleccione el StackSet en el que desea llevar a cabo la detección de desviaciones.
3. En el menú Actions (Acciones) seleccione Detect drifts (Detectar desviaciones).

CloudFormation muestra una barra de información que indica que se ha iniciado la detección de desviaciones en el StackSet seleccionado.

4. Opcional: para monitorear el progreso de la operación de detección de desviaciones:
 - a. Seleccione el nombre del StackSet para mostrar la página Detalles de StackSet.
 - b. Seleccione la ficha Operations (Operaciones) seleccione la operación de detección de desviaciones y, a continuación, seleccione View drift details (Ver detalles de desviación).

CloudFormation muestra el cuadro de diálogo Operation details (Detalles de la operación).

5. Espere hasta que CloudFormation complete la operación de detección de desviaciones. Cuando la operación de detección de desviaciones finaliza, CloudFormation actualiza los valores de Estado de desviación y Hora de la última comprobación de desviaciones del StackSet. Estos campos se muestran en la pestaña Información general de la página Detalles de StackSet para el StackSet seleccionado.

La operación de detección de desviaciones puede tardar un poco, según del número de instancias de pila incluidas en el StackSet, así como del número de recursos incluidos en él. Solo puede ejecutar una única operación de detección de desviaciones en un StackSet determinado a la vez. CloudFormation continúa con la operación de detección de desviaciones incluso después de cerrar la barra de información.

6. Para revisar los resultados de la detección de desviaciones para las instancias de pila de un StackSet, seleccione la pestaña Instancias de pila.

La columna Stack name (Nombre de pila) muestra el nombre de la pila asociado a cada instancia de pila, y la columna Drift status (Estado de desviación) muestra el estado de desviación de dicha pila. Se considera que una pila se ha desviado si uno o varios de sus recursos se han desviado.

7. Para revisar los resultados de la detección de desviaciones de la pila asociada a una instancia de pila específica:
 - a. Seleccione la pestaña Operaciones.
 - b. Seleccione la operación de desviación de la que quiere ver los resultados de la detección de desviación. Un panel dividido mostrará el estado de la instancia de pila y el motivo de la operación seleccionada. Para una operación de desviación, la columna de razón de estado muestra el estado de desviación de una instancia de pila.
 - c. Seleccione la instancia de pila de la que quiere ver los detalles de desviación y elija Ver desviaciones de recursos. En la tabla Estado de desviación de los recursos de la página Desviaciones de recursos, se muestra cada recurso de la pila con su estado de desviación y la última vez que se inició la detección de desviaciones en el recurso. El ID lógico y el ID físico de cada recurso se muestra para ayudarle a identificarlos.
8. Puede ordenar los recursos en función de su estado mediante la columna Drift status (Estado de desviación).

Para ver los detalles de un recurso modificado:

- Con el recurso seleccionado, seleccione Ver detalles de desviación.

CloudFormation muestra la página de detalles de desviación para ese recurso en particular. En esta página se detallan las diferencias del recurso. También muestra los valores de propiedades previstos y actuales del recurso.

 Note

Si la pila pertenece a una región y una cuenta diferentes a las que está conectado en ese momento, el botón Detectar desviación se deshabilitará y no podrá ver los detalles.

Detección de la desviación en un StackSet (AWS CLI)

Para detectar la desviación en una pila completa a través de la AWS CLI, utilice el siguiente procedimiento:

Detección de la desviación en un StackSet

1. Utilice el comando [detect-stack-set-drift](#) para detectar la desviación en un StackSet completo y sus instancias de pila asociadas.

En el ejemplo siguiente se inicia la detección de desviación en el StackSet `stack-set-drift-example`.

```
aws cloudformation detect-stack-set-drift \  
  --stack-set-name stack-set-drift-example
```

Salida:

```
{  
  "OperationId": "c36e44aa-3a83-411a-b503-cb611example"  
}
```

2. Debido a que las operaciones de detección de desviaciones del StackSet pueden ser operaciones de larga duración, utilice el comando [describe-stack-set-operation](#) para supervisar

el estado de la operación de desviaciones. Este comando toma el ID de operación del StackSet que el comando `detect-stack-set-drift` devuelve.

En los ejemplos siguientes se utiliza el ID de operación del ejemplo anterior para devolver información sobre la operación de detección de desviación del StackSet. En este ejemplo, la operación todavía se está ejecutando. De las siete instancias de pila asociadas a este StackSet, ya se detectó la desviación de una instancia de pila, dos instancias están sincronizadas y la detección de desviaciones de las cuatro instancias de pila restantes aún está en curso. Dado que una instancia se ha desviado, el estado de desviación del StackSet en sí es ahora **DRIFTED**.

```
aws cloudformation describe-stack-set-operation \  
  --stack-set-name stack-set-drift-example \  
  --operation-id c36e44aa-3a83-411a-b503-cb611example
```

Salida:

```
{  
  "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"  
  }  
}
```

```
}
```

Al ejecutar el mismo comando más adelante, este ejemplo muestra la información devuelta una vez completada la operación de detección de desviación. Dos de las siete instancias totales de pila asociadas a este StackSet se han desviado, lo que representa el estado de desviación del propio StackSet como DRIFTED.

```
aws cloudformation describe-stack-set-operation \  
  --stack-set-name stack-set-drift-example \  
  --operation-id c36e44aa-3a83-411a-b503-cb611example
```

Salida:

```
{  
  "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. Una vez finalizada la operación de detección de desviaciones del StackSet, use los comandos `describe-stack-set`, `list-stack-instances`, `describe-stack-instance` y `list-stack-instance-resource-drifts` para revisar los resultados.

El comando [describe-stack-set](#) incluye la misma información detallada de desviación devuelta por el comando `describe-stack-set-operation`.

```
aws cloudformation describe-stack-set \  
  --stack-set-name stack-set-drift-example
```

Salida:

```
{  
  "StackSet": {  
    "Status": "ACTIVE",  
    "Description": "Demonstration of drift detection on stack sets.",  
    "Parameters": [],  
    "Tags": [  
      {  
        "Value": "Drift detection",  
        "Key": "Feature"  
      }  
    ],  
    "ExecutionRoleName": "AWSCloudFormationStackSetExecutionRole",  
    "Capabilities": [],  
    "AdministrationRoleARN": "arn:aws:iam::123456789012:role/  
AWSCloudFormationStackSetAdministrationRole",  
    "StackSetDriftDetectionDetails": {  
      "DriftedStackInstancesCount": 2,  
      "TotalStackInstancesCount": 7,  
      "LastDriftCheckTimestamp": "2019-12-04T20:36:55.612Z",  
      "InProgressStackInstancesCount": 0,  
      "DriftStatus": "DRIFTED",  
      "DriftDetectionStatus": "COMPLETED",  
      "InSyncStackInstancesCount": 5,  
      "FailedStackInstancesCount": 0  
    },  
    "StackSetARN": "arn:aws:cloudformation:us-east-1:123456789012:stackset/  
stack-set-drift-example:bd1f4017-d4f9-432e-a73f-8c22example",  
    "TemplateBody": [details omitted],  
    "StackSetId": "stack-set-drift-example:bd1f4017-d4f9-432e-  
a73f-8c22ebexample",  
  }  
}
```

```

    "StackSetName": "stack-set-drift-example"
  }
}

```

Puede utilizar el comando [list-stack-instances](#) para devolver información de resumen sobre las instancias de pila asociadas a un StackSet, incluido el estado de desviación de cada instancia de pila.

En este ejemplo, la ejecución de `list-stack-instances` en el StackSet de ejemplo con el filtro de estado de desviación establecido en `DRIFTED` permite identificar qué dos instancias de pila tienen el estado de desviación `DRIFTED`.

```

aws cloudformation list-stack-instances \
  --stack-set-name stack-set-drift-example \
  --filters Name=DRIFT_STATUS,Values=DRIFTED

```

Salida:

```

{
  "Summaries": [
    {
      "StackId": "arn:aws:cloudformation:eu-west-1:123456789012:stack/StackSet-stack-set-drift-example-b0fb6083-60c0-4e39-af15-2f071e0db90c/0e4f0940-16d4-11ea-93d8-0641cexample",
      "Status": "CURRENT",
      "Account": "012345678910",
      "Region": "eu-west-1",
      "LastDriftCheckTimestamp": "2019-12-04T20:37:32.687Z",
      "DriftStatus": "DRIFTED",
      "StackSetId": "stack-set-drift-example:bd1f4017-d4f9-432e-a73f-8c22eexample",
      "LastOperationId": "c36e44aa-3a83-411a-b503-cb611example"
    },
    {
      "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/StackSet-stack-set-drift-example-b7fde68e-e541-44c2-b33d-ef2e2988071a/008e6030-16d4-11ea-8090-12f89example",
      "Status": "CURRENT",
      "Account": "123456789012",
      "Region": "us-east-1",
      "LastDriftCheckTimestamp": "2019-12-04T20:34:28.275Z",
      "DriftStatus": "DRIFTED",

```

```

        "StackSetId": "stack-set-drift-example:bd1f4017-d4f9-432e-
a73f-8c22eexample",
        "LastOperationId": "c36e44aa-3a83-411a-b503-cb611example"
    },

    [additional stack instances omitted]

]
}

```

El comando [describe-stack-instance](#) también devuelve esta información, pero para una sola instancia de pila, como en el ejemplo siguiente.

```

aws cloudformation describe-stack-instance \
  --stack-set-name stack-set-drift-example \
  --stack-instance-account 012345678910 --stack-instance-region us-east-1

```

Salida:

```

{
  "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"
  }
}

```

- Una vez que haya identificado qué instancias de pila se han desviado, puede utilizar la información sobre las instancias de pila devueltas por los comandos `list-stack-instances` o `describe-stack-instance` para ejecutar el comando [list-stack-instance-resource-drifts](#). Este comando devuelve información detallada sobre qué recursos de la pila se desviaron para una operación de desvío concreta.

En el siguiente ejemplo se usa el parámetro `--stack-instance-resource-drift-statuses` para solicitar información de desviación de pila para los recursos que se modificaron o eliminaron en el ejemplo de operación de desviación anterior. La solicitud devuelve información sobre un recurso que se modificó, incluidos detalles sobre dos de sus propiedades y sus valores modificados. No se ha eliminado ningún recurso.

```
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
```

Salida:

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

```
]
}
```

Detención de la detección de desviaciones en un StackSet

Dado que la detección de desviaciones en un StackSet puede ser una operación de larga duración, puede haber instancias en las que desee detener una operación de detección de desviaciones que se esté ejecutando en un StackSet.

Detención de la detección de desviaciones en un StackSet (consola)

1. Abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation>.
2. En la página StackSets, seleccione el nombre del StackSet.

CloudFormation muestra la página Detalles de StackSet para el StackSet seleccionado.

3. En la página StackSets details (Detalles de conjuntos de pilas) seleccione la ficha Operations (Operaciones) y, a continuación, seleccione la operación de detección de desviaciones.
4. Seleccione Stop operation (Detener operación).

Detención de la detección de desviaciones en un StackSet (AWS CLI)

- Utilice el comando [stop-stack-set-operation](#). Debe proporcionar tanto el nombre del StackSet como el ID de la operación del StackSet de detección de desviaciones.

```
aws cloudformation stop-stack-set-operation \
  --stack-set-name stack-set-drift-example \
  --operation-id 624af370-311a-11e8-b6b7-500cexample
```

Importación de pilas en CloudFormation StackSets

Una operación de importación de pilas puede importar pilas existentes en StackSets nuevos o existentes, de modo que puede migrar las pilas existentes a un StackSet en una sola operación.

Para StackSets autoadministrado, la operación de importación puede importar pilas en la cuenta de administrador o en diferentes cuentas de destino y Regiones de AWS. Para los conjuntos de pilas administradas por servicios, la operación de importación puede importar cualquier pila en la misma AWS Organizations como una cuenta de administración.

A continuación se presentan las consideraciones y limitaciones a la hora de importar pilas a StackSets:

- La operación de importación puede importar hasta 10 pilas con los ID de pila en línea o hasta 200 pilas mediante un objeto de Amazon S3.
- No se admite la propiedad NoEcho. Las pilas que contienen NoEcho no se importarán en los nuevos StackSets con la importación de StackSets.
- Las pilas solo pueden pertenecer a un StackSet.
- Puede implementar etiquetas de pila en el StackSet especificando etiquetas explícitamente como parámetros en la operación de importación de pilas.
- Las anulaciones de parámetros personalizados de una pila no se ven afectadas durante la operación de importación.
- Los parámetros de las instancias de pila importadas no se pueden actualizar con la opción Editar detalles de StackSet. Debe utilizar la opción Anular parámetros de StackSet. Para obtener más información sobre los parámetros de anulación, consulte [Anulación de parámetros en pilas](#).
- Las cuotas del conjunto de pilas y las instancias de pilas se aplican al momento de importar las pilas. Para obtener más información sobre las cuotas, consulte [Comprensión de las cuotas de CloudFormation](#).

Temas

- [Importación de las pilas administradas automáticamente para conjuntos de pilas de CloudFormation](#)
- [Importación de las pilas administradas por servicios para conjuntos de pilas de CloudFormation](#)
- [Reversión de las importaciones de pilas a CloudFormation StackSets](#)

Importación de las pilas administradas automáticamente para conjuntos de pilas de CloudFormation

La operación de importación de pilas de CloudFormation puede importar pilas existentes en StackSets nuevos o existentes, de modo que pueda migrar las pilas existentes a un StackSet en una sola operación. Al utilizar la importación de pilas, se evita el tiempo de inactividad y las interrupciones sin eliminar ni volver a crear esos recursos. Una vez que la pila se importe en un StackSet, la pila original se convertirá en una instancia de pila del conjunto de pila especificado.

Consideraciones sobre las importaciones de pilas autoadministradas

- La operación de importación de pilas requiere una cuenta de administrador en la que se cree un StackSet y una cuenta de destino que contenga una pila.
- La cuenta de destino debe tener permisos para utilizar la operación `GetTemplate` con la entrada de ID de pila o ARN. Por eso, su cuenta de administrador debe disponer de los permisos `AWSCloudFormationStackSetAdministrationRole` o `AWSCloudFormationStackSetsExecutionRole`.

Temas

- [Importación de una pila existente en un nuevo StackSet \(consola\)](#)
- [Importación de una pila existente en un StackSet existente \(consola\)](#)
- [Importación de una pila en un StackSet \(AWS CLI\)](#)

Importación de una pila existente en un nuevo StackSet (consola)

Antes de empezar, identifique la pila que desee importar.

1. Inicie sesión en la Consola de administración de AWS y abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation>.
2. En el panel de navegación, seleccione StackSets (Conjuntos de pilas).
3. En la parte superior de la página StackSets (Conjuntos de pilas), seleccione Create StackSet (Crear conjunto de pilas).
4. En la página Elegir una plantilla, especifique una plantilla utilizando una de las siguientes opciones y seleccione Siguiente.
 - Elija Amazon S3 URL (URL de Amazon S3) y especifique la URL de la plantilla en el cuadro de texto.
 - Elija Upload a template file (Cargar un archivo de plantilla) y busque la plantilla.
 - Elija From stack ID (Desde el ID de la pila) e ingrese el ID de la pila.
5. En la página Especificar detalles del StackSet, ingrese el nombre del StackSet que desee crear y seleccione Siguiente.

(Opcional) Ingrese una descripción del StackSet.

6. En la página Configurar las opciones del StackSet, revise las opciones y seleccione Siguiente.

7. En la página Establecer opciones de implementación, seleccione Importar pilas al conjunto de pilas.
8. Ingrese el ID de la pila que desea importar en el campo Stacks to import (Pilas para importar). Por ejemplo, *arn:aws:cloudformation:us-east-1:123456789012:stack/StackToImport/f449b250-b969-11e0-a185-5081d0136786*.

(Opcional) Seleccione Agregar otro ID de pila e ingrese el ID de otra pila que desee importar. Puede agregar hasta 10 pilas por cada operación de importación de pilas.

9. Revise las opciones de implementación y seleccione Siguiente.
10. En la página Revisar, revise sus opciones y las propiedades del StackSet. Cuando lo tenga todo listo para importar la pila en su StackSet, seleccione Enviar.

Resultados: La pila importada ahora es una instancia de pila del StackSet especificado. Para obtener más información sobre el estado de importación de la pila, consulte [Códigos de estado de StackSets](#).

Importación de una pila existente en un StackSet existente (consola)

Antes de empezar, identifique la pila que desee importar.

1. Inicie sesión en la Consola de administración de AWS y abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation>.
2. En el panel de navegación, seleccione StackSets (Conjuntos de pilas).
3. En la página StackSets, seleccione el StackSet en el que desee importar una pila.
4. Con el StackSet seleccionado, elija Agregar pilas al StackSet en el menú Acciones.
5. En la página Establecer opciones de implementación, seleccione Importar pilas al conjunto de pilas e ingrese el ID de la pila que desea importar en el campo Pilas para importar. Por ejemplo, *arn:aws:cloudformation:us-east-1:123456789012:stack/StackToImport/f449b250-b969-11e0-a185-5081d0136786*.

(Opcional) Seleccione Agregar otro ID de pila e ingrese el ID de otra pila que desee importar. Puede agregar hasta 10 pilas por cada operación de importación de pilas.

6. Elija Siguiente.
7. En la página Especificar anulaciones, revise las opciones y seleccione Siguiente.
8. En la página Revisar, revise sus opciones y las propiedades del StackSet. Cuando lo tenga todo listo para crear su StackSet, elija Enviar.

Resultados: La pila importada ahora es una instancia de pila del StackSet especificado. Para obtener más información sobre el estado de importación de la pila, consulte [Códigos de estado de StackSets](#).

Importación de una pila en un StackSet (AWS CLI)

Importación de una pila existente en un nuevo StackSet

El comando `create-stack-set` siguiente crea un StackSet e importa la pila especificada. La pila que desea importar se identifica mediante su ARN. Sustituya el texto del marcador de posición por su propia información.

```
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
```

Importación de una pila existente en un StackSet existente

El comando `import-stacks-to-stack-sets` siguiente importa la pila especificada en el StackSet *MyStackSet*. La pila que desea importar se identifica mediante su ARN. Sustituya el texto del marcador de posición por su propia información.

```
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
```

Para especificar más de una pila, utilice el siguiente formato para el valor de la opción `--stack-ids`.

```
--stack-ids "arn_1" "arn_2"
```

Para clonar la pila importada en otras regiones y cuentas

El comando `create-stack-instances` siguiente agrega instancias de pila al StackSet. Sustituya el texto del marcador de posición por su propia información.

```
aws cloudformation create-stack-instances \  
  --stack-set-name MyStackSet \  
  --accounts '["account_ID_1", "account_ID_2"]' \  
  --regions '["region_1", "region_2"]'
```

Importación de las pilas administradas por servicios para conjuntos de pilas de CloudFormation

La operación de importación de pilas de CloudFormation puede importar pilas existentes en StackSets nuevos o existentes, de modo que pueda migrar las pilas existentes a un StackSet en una sola operación. StackSets amplía la funcionalidad de las pilas para que pueda crear, actualizar o eliminar pilas de varias cuentas y regiones con una sola operación.

Consideraciones para la importación de pilas administradas por servicios

- La operación de importación de la pila requiere una cuenta de administración o una cuenta de administrador delegado en la que pueda administrar los AWS Organizations asociados; por ejemplo, para permitir el acceso de confianza con StackSets.
- Las cuentas de destino deben ser miembros de la AWS Organizations administrada por la cuenta de administración o la cuenta de administrador delegado.
- La pila de destino existe en una de las unidades organizativas de destino.
- La cuenta de destino debe ser miembro de AWS Organizations.
- El acceso a AWS Organizations debe tener el estado ACTIVATED para las organizaciones.
- Las pilas que se van a importar deben estar presentes en cualquiera de las cuentas de los miembros y no en la cuenta de administración.

Temas

- [Importación de una pila administrada por servicios en un nuevo StackSet \(consola\)](#)
- [Creación e importación de una pila administrada por servicios en un StackSet existente \(consola\)](#)
- [Importación de una pila administrada por servicios en un StackSet existente \(consola\)](#)
- [Importación de una pila administrada por servicios en un StackSet \(AWS CLI\)](#)

Importación de una pila administrada por servicios en un nuevo StackSet (consola)

Importación de una pila en un nuevo StackSet mediante la Consola de administración de AWS

Para importar una pila nueva en un StackSet, identifique una pila que contenga el recurso que desee importar.

1. Inicie sesión en la Consola de administración de AWS y abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation>.
2. En el panel de navegación, seleccione StackSets (Conjuntos de pilas).
3. En la parte superior de la página StackSets (Conjuntos de pilas), seleccione Create StackSet (Crear conjunto de pilas).
4. En la página Elegir una plantilla, haga lo siguiente:
 - a. En StackSet permission model, seleccione Permisos administrados por servicios.
 - b. En Requisito previo: preparar plantilla, seleccione La plantilla está lista y elija la plantilla mediante una de las siguientes opciones:
 - En URL de Amazon S3, ingrese su URL de Amazon S3 en el campo URL de Amazon S3.
 - En Cargar un archivo de plantilla, seleccione una plantilla de CloudFormation de su equipo local.

Acepte la configuración y elija Next (Siguiente).

5. En la página Especificar detalles del StackSet, haga lo siguiente:
 - a. Ingrese un nombre de StackSet en el cuadro Nombre del StackSet.
 - b. (Opcional) Ingrese una descripción en la sección StackSet description (Descripción de StackSet).

En la página Configurar las opciones del StackSet, revise las opciones y seleccione Siguiente.

6. En la página Establecer opciones de implementación, realice las siguientes acciones:
 - a. En Agregar pilas al conjunto de pilas, seleccione Importar pilas al conjunto de pilas.
 - b. En Stacks to import (Pilas para importar), elija el método de importación de pila.
 - i. En Stack ID (ID de la pila), ingrese el ID de la pila.
 - ii. En URL de la pila ingrese la URL de Amazon S3.
7. En Asociar unidades organizativas, haga lo siguiente:
 - a. Seleccione Asociar con la organización para utilizar la unidad organizativa raíz.

- b. Seleccione Asociar con unidades organizativas (OU) para ingresar los ID de las unidades organizativas principales para las pilas que se van a importar. Por ejemplo, si Stack 1 y Stack 2 están debajo de OU1, y Stack 3 está debajo de OU2, ingrese OU1 y OU2.

Acepte la configuración y elija Next (Siguiente).

8. Revise las opciones en la página Review (Revisar) y elija Submit (Enviar).

Creación e importación de una pila administrada por servicios en un StackSet existente (consola)

Para importar una pila existente en un nuevo StackSet, identifique una pila que contenga el recurso que desee importar.

Creación de un StackSet e importación de una pila

1. Inicie sesión en la Consola de administración de AWS y abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation>.
2. En el panel de navegación, seleccione StackSets (Conjuntos de pilas).
3. En la parte superior de la página StackSets (Conjuntos de pilas), seleccione Create StackSet (Crear conjunto de pilas).
4. En la página Elegir una plantilla, haga lo siguiente:
 - a. En StackSet permission model, seleccione Permisos administrados por servicios.
 - b. En Requisito previo: preparar plantilla, seleccione La plantilla está lista y elija la plantilla mediante una de las siguientes opciones:
 - En URL de Amazon S3, ingrese su URL de Amazon S3 en el campo URL de Amazon S3.
 - En Cargar un archivo de plantilla, seleccione una plantilla de CloudFormation de su equipo local.

Acepte la configuración y elija Next (Siguiente).

5. En la página Especificar detalles del StackSet, haga lo siguiente:
 - a. Ingrese un nombre de StackSet en el cuadro Nombre del StackSet.

- b. (Opcional) Ingrese una descripción en la sección StackSet description (Descripción de StackSet).

En la página Configurar las opciones del StackSet, revise las opciones y seleccione Siguiente.

6. En la página Establecer opciones de implementación, realice las siguientes acciones:
 - En Add stacks to stack set (Agregar pilas al conjunto de pilas), seleccione Deploy new stacks (Implementar pilas nuevas).
7. En la sección Asociar unidades organizativas, haga lo siguiente:
 - a. Seleccione Asociar con la organización para utilizar la unidad organizativa raíz.
 - b. Seleccione Asociar con unidades organizativas (OU) para ingresar los ID de las unidades organizativas principales para las pilas que se van a importar. Por ejemplo, si Stack 1 y Stack 2 están debajo de OU1, y Stack 3 está debajo de OU2, ingrese OU1y OU2.
8. En Specify regions (Especificar regiones) y Deployment options (Opciones de implementación), revise las opciones.

Acepte la configuración y elija Next (Siguiente).

9. Revise las opciones en la página Review (Revisar) y elija Submit (Enviar).

Importación de una pila administrada por servicios en un StackSet existente (consola)

Seleccione su StackSet e identifique la pila que desee importar.

Importación de una pila en un StackSet existente

1. Inicie sesión en la Consola de administración de AWS y abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation>.
2. En el panel de navegación, seleccione StackSets (Conjuntos de pilas).
3. Seleccione el StackSet en el que desee importar una pila y, a continuación, elija Agregar pilas al StackSet en el menú desplegable Acciones.
4. En la página Establecer opciones de implementación, realice las siguientes acciones:
 - a. En Agregar pilas al conjunto de pilas, seleccione Importar pilas al conjunto de pilas.
 - b. En Pilas para importar, haga lo siguiente

- i. En Stack ID (ID de la pila), ingrese el ID de la pila.
- ii. En URL de la pila, ingrese la URL de Amazon S3.
- c. En Asociar unidades organizativas, haga lo siguiente:
 - i. Seleccione Asociar con la organización para utilizar la unidad organizativa raíz.
 - ii. Seleccione Asociar con unidades organizativas (OU) para ingresar los ID de las unidades organizativas principales para las pilas que se van a importar. Por ejemplo, si Stack 1 y Stack 2 están debajo de OU1, y Stack 3 está debajo de OU2, ingrese OU1y OU2.

Acepte la configuración y elija Next (Siguiente).

5. Revise la página Specify overrides (Especificar anulaciones) y elija Next (Siguiente).
6. Confirme y revise la página Review (Revisar) y elija Submit (Enviar).

Importación de una pila administrada por servicios en un StackSet (AWS CLI)

Una vez creado el StackSet, puede importar las pilas si pasa el ID de las pilas que se van a importar. También puede pasar la lista de ID de unidades organizativas a las que desea asignar el conjunto.

CloudFormation importará las pilas proporcionadas por el usuario dentro de esas UO y utilizará dichas UO como destinos de implementación para el StackSet. Los ID de pila que se muestran en la entrada se asignarán internamente a la unidad organizativa más próxima en la lista de ID de unidades organizativas. Si una pila no pertenece a un ID de unidad organizativa existente en la lista de entrada, la AWS CLI devolverá el error `StackNotFoundException`.

La operación `import-stacks-to-stack-set` crea instancias de pila para las pilas en la entrada de ID de unidad organizativa. En los siguientes ejemplos de la AWS CLI, se utiliza la operación `import-stacks-to-stack-set` para importar una pila en un StackSet.

- Para utilizar la operación `import-stacks-to-stack-sets`, especifique los `stack-ids` o `stack-ids-url` que desee importar a su conjunto de pilas.

```
aws cloudformation import-stacks-to-stack-set \  
  --stack-set-name ServiceMangedStackSet \  
  --stack-ids "arn:123456789012:us-east-1:Stack1" \  
  --organizational-unit-ids ou-examplerootid111-exampleouid111
```



```
aws cloudformation import-stacks-to-stack-set \  
  --stack-set-name 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

La operación `import-stacks-to-stack-sets` requiere que especifique al menos un ID de unidad organizativa para que pueda asociar la pila que se va a importar a dicha unidad organizativa en particular. Esta operación no crea instancias de pila para otras cuentas de miembros en las unidades organizativas asociadas. Para actualizar las cuentas de miembros de las unidades organizativas asociadas, utilice `create-stack-instances` o `update-stack-instances`.

`create-stack-set` crea instancias de pila para todas las cuentas de las unidades organizativas con una plantilla brindada por el usuario, ya sea mediante carga directa o Amazon S3. En los siguientes ejemplos de la AWS CLI, se utiliza la operación `create-stack-set` para importar una pila en un nuevo StackSet.

- Para utilizar la operación `create-stack-set`, especifique el nombre del StackSet e importe una pila en un StackSet recién creado.

```
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
```

Reversión de las importaciones de pilas a CloudFormation StackSets

Si tiene cambios no deseados en su instancia de pila, puede revertir la importación de la instancia de pila.

Al revertir la importación de una instancia de pila, CloudFormation elimina la instancia de pila del StackSet, pero conserva los recursos de la pila.

Para revertir una operación de importación de pila, realice el siguiente procedimiento.

1. Especifique un atributo `DeletionPolicy` de `Retain` para cada recurso que desee conservar después de eliminar la instancia de la pila. Para obtener más información, consulte [Reversión de una operación de importación](#).
2. Elimine las instancias de pila del `StackSet`. Para obtener más información, consulte [Eliminación de pilas de CloudFormation StackSets](#).
3. Elimine su `StackSet`. Para obtener más información, consulte [Eliminación de StackSets de CloudFormation](#).

Prácticas recomendadas para utilizar CloudFormation StackSets

En esta sección se describen las prácticas recomendadas para definir una plantilla de `StackSet`, crear o agregar pilas a un `StackSet` o actualizar un `StackSet`.

Si es la primera vez que usa CloudFormation, consulte el tema [Prácticas recomendadas de CloudFormation](#) para ver más recomendaciones que pueden ayudarlo a usar CloudFormation de forma más eficaz y segura.

Temas

- [Definición de la plantilla](#)
- [Cómo crear o agregar pilas al StackSet](#)
- [Actualización de pilas en un StackSet](#)

Definición de la plantilla

- Defina la plantilla que desee estandarizar en varias cuentas, dentro de varias regiones.
- Al crear la plantilla, asegúrese de que los recursos globales (como roles de IAM y buckets de Amazon S3) no tengan conflictos de nomenclatura al crearse en más de una región de la misma cuenta.
- Un `StackSet` tiene una plantilla y un conjunto de parámetros únicos. La misma pila se crea en todas las cuentas que están asociadas a un `StackSet`. Al crear las plantillas, hágalas lo suficientemente granulares para permitir un buen equilibrio entre el control y la estandarización.
- Le recomendamos que almacene su plantilla en un bucket de Amazon S3.

Cómo crear o agregar pilas al StackSet

- Compruebe que la adición de las instancias de pila al StackSet inicial funcione antes de agregar un mayor número de instancias de pila al StackSet.
- Elija las opciones de implementación (despliegue) que funcionen para su caso de uso.
 - Para obtener una implementación más conservadora, establezca Maximum Concurrent Accounts (Número máximo de cuentas simultáneas) en 1 y Failure Tolerance (Tolerancia a errores) en 0. Configure la región de menor impacto en primer lugar en la lista Region Order (Orden de regiones). Comience con una región.
 - Para una implementación más rápida, aumente los valores de tolerancia a Maximum Concurrent Accounts (Número máximo de cuentas simultáneas) y Failure Tolerance (Tolerancia a errores) según sea necesario.
- Las operaciones en StackSets dependen de la cantidad implicada de instancias de pila y puede tardar bastante tiempo.

Actualización de pilas en un StackSet

- De forma predeterminada, la actualización de un StackSet actualiza todas las instancias de pila. Si dispone de 20 cuentas en dos regiones, tendrá 40 instancias de pila y todas se actualizarán cuando actualice el StackSet.

Para StackSets con un gran número de instancias de pila, le recomendamos que actualice selectivamente las instancias de pila en unas cuantas cuentas de prueba antes de actualizar todas las instancias de pila para probar la versión actualizada de una plantilla.

- Para obtener un control más minucioso de la actualización de pilas individuales dentro del StackSet, considere la posibilidad de crear varios StackSets.
- Actualizar un StackSet que contenga un gran número de pilas puede tardar bastante tiempo. En esta versión, solo se permite una operación a la vez en un StackSet. Planifique sus actualizaciones para que no se le impida llevar a cabo otras operaciones en el StackSet.

Plantilla de ejemplo de CloudFormation StackSets

En esta sección se incluyen enlaces a algunas plantillas de CloudFormation de ejemplo que pueden ayudarlo a usar CloudFormation StackSets en su empresa. Las plantillas que figuran en esta sección habilitan [AWS CloudTrail](#) o [AWS Config](#) y las reglas en su interior.

⚠ Important

Como práctica recomendada de seguridad a la hora de permitir el acceso de AWS Config a un bucket de Amazon S3, le recomendamos encarecidamente que restrinja el acceso a la política de bucket con la condición `AWS:SourceAccount`. Las plantillas nuevas se actualizan para tener `AWS:SourceAccount`. Si su política de bucket actual no sigue esta práctica recomendada de seguridad, le recomendamos encarecidamente que edite esa política de bucket para incluir esta protección. Esto garantiza que solo AWS Config reciba acceso en nombre de los usuarios esperados.

Descripción	Enlace de S3
Habilitar AWS CloudTrail	https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/EnableAWSCloudtrail.yml
Habilitar AWS Config	https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/EnableAWSConfig.yml
Habilitar AWS Config con registro central	https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/EnableAWSConfigForOrganizations.yml
Habilitar las políticas predeterminadas de Amazon Data Lifecycle Manager en toda una organización de AWS o en cuentas de AWS específicas	- Política predeterminada para las instantáneas de EBS: https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/DataLifecycleManagerEBSSnapshotDefaultPolicy.yml - Política predeterminada para las AMI basadas en EBS: https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/DataLifecycleManagerAMIDefaultPolicy.yml

Descripción	Enlace de S3
	https://s3.amazonaws.com/cloudformation-stacks/sample-templates-us-east-1/ConfigRuleEncryptedVolumes.yml
Configurar una regla de AWS Config para determinar la habilitación de CloudTrail	https://s3.amazonaws.com/cloudformation-stacks/sample-templates-us-east-1/ConfigRuleCloudTrailEnabled.yml
Configurar una regla de AWS Config para determinar la habilitación de FMA raíz	https://s3.amazonaws.com/cloudformation-stacks/sample-templates-us-east-1/ConfigRuleRootAccountMFAEnabled.yml
Configurar una regla de AWS Config para determinar si las EIP están conectadas	https://s3.amazonaws.com/cloudformation-stacks/sample-templates-us-east-1/ConfigRuleEipAttached.yml
Configurar una regla de AWS Config para determinar si los volúmenes de EBS están cifrados	https://s3.amazonaws.com/cloudformation-stacks/sample-templates-us-east-1/ConfigRuleEncryptedVolumes.yml

Solución de problemas de CloudFormation StackSets

Este tema contiene algunos problemas habituales relacionados con StackSets y las soluciones sugeridas.

Temas

- [Motivos comunes para el fallo en una operación de la pila](#)
- [Reintento de operaciones fallidas de creación o actualización de pilas](#)
- [No se puede eliminar una instancia de pila a causa de un error](#)
- [Error en la operación de importación de pila](#)
- [Recuento de fallos de instancias de pila para las operaciones de StackSets](#)

Motivos comunes para el fallo en una operación de la pila

Problema: se produjo un error en una operación de la pila, y el estado de la instancia de la pila es OUTDATED.

Causa: puede haber varias causas comunes para el error en la operación de la pila.

- Permisos insuficientes en una cuenta de destino para la creación de recursos que se especifican en la plantilla.
- La plantilla de CloudFormation podría tener errores. Valide la plantilla en CloudFormation y arregle los errores antes de intentar crear el StackSet.
- La plantilla podría intentar crear recursos globales que debe ser únicos pero no lo son, como los buckets de S3.
- Un número de cuenta de destino especificado no existe. Compruebe los números de cuenta de destino que ha especificado en la página Set deployment options (Establecer opciones de implementación) del asistente.
- La cuenta de administrador no tiene una relación de confianza con la cuenta de destino.
- El número máximo de un recurso que se especifica en la plantilla ya existe en su cuenta de destino. Por ejemplo, es posible que haya alcanzado el límite de roles de IAM permitidos en una cuenta de destino, pero la plantilla crea más roles de IAM.
- Ha alcanzado el número máximo de pilas permitidas en un StackSet. Para ver el número máximo de pilas por StackSet, consulte [Comprensión de las cuotas de CloudFormation](#).

Solución: Para obtener más información acerca de los permisos necesarios de las cuentas de administrador y de destino antes de poder crear StackSets, consulte [Concesión de permisos a todos los usuarios de la cuenta de administrador para administrar pilas en todas las cuentas de destino](#).

Reintento de operaciones fallidas de creación o actualización de pilas

Problema: se produjo un error al crear o actualizar una pila, y el estado de la instancia de la pila es OUTDATED. Para solucionar el error de creación o actualización de una pila, abra la consola de CloudFormation y vea los eventos de la pila, que tendrán el estado DELETED (para las operaciones de creación con errores) o FAILED (para las operaciones de actualización con errores). Examine los eventos de pila y busque la columna Status reason (Motivo del estado). El valor de Status reason (Motivo del estado) explica por qué se produjo un error en la operación de la pila.

Una vez corregida la causa del fallo de creación de la pila, y cuando esté listo para reintentar la creación de la pila, siga estos pasos.

Solución: realice los siguientes pasos para reintentar la operación de la pila.

1. En la consola, seleccione el StackSet que contiene la pila en la que la operación ha fallado.
2. En el menú Actions (Acciones), elija Edit StackSet details (Editar detalles de StackSet) para volver a intentar la creación o actualización de pilas.
3. En la página Especificar plantilla, para utilizar la misma plantilla de CloudFormation, mantenga la opción predeterminada (Utilizar la plantilla actual). Si la operación de la pila ha generado un error porque la plantilla requería cambios, y desea cargar una plantilla revisada, elija Upload a template to Amazon S3 (Cargar una plantilla en Amazon S3) en su lugar y, a continuación, elija Browse (Explorar) para seleccionar la plantilla actualizada. Cuando haya terminado de cargar la plantilla revisada, elija Next (Siguiente).
4. En la página Especificar los detalles de la pila, si no va a cambiar ninguno de los parámetros que son específicos de la plantilla, elija Siguiente.
5. En la página Set deployment options (Establecer opciones de implementación), cambie los valores predeterminados de Maximum concurrent accounts (Número máximo de cuentas simultáneas) y Failure tolerance (Tolerancia a errores), si lo desea. Para obtener más información sobre estas opciones, consulte [Opciones de operaciones con StackSets](#).
6. En la página Review (Revisar), revise la selección y marque la casilla para reconocer las capacidades de IAM. Elija Enviar.
7. Si la pila no se actualiza correctamente, repita este procedimiento, después de haber resuelto todos los problemas que impiden la creación de la pila.

No se puede eliminar una instancia de pila a causa de un error

Problema: se produce un error que evita la eliminación de una instancia de pila.

Causa: ninguna pila con protección de terminación de pilas habilitada se podrá eliminar.

Solución: determine si se ha habilitado la protección de terminación en la pila. Si está habilitada, deshabilítela y, a continuación, intente realizar la operación de eliminación de la instancia de pila de nuevo.

Error en la operación de importación de pila

Problema: Una operación de importación de pilas no permite importar pilas existentes en StackSets nuevos o existentes. El estado de la instancia de pila es INOPERABLE.

Solución: para restaurar la operación de importación de la pila, lleve a cabo las siguientes tareas.

1. Use la opción Delete Stacks from StackSets y habilite RetainStacks durante la configuración y, a continuación, proceda a eliminar las instancias de pila del StackSet. Para obtener más información, consulte [Eliminación de pilas de CloudFormation StackSets](#).
2. Verá que la instancia de pila del StackSet se actualiza para eliminar la instancia de pila INOPERABLE.
3. Corrija las instancias de la pila según el error de importación y vuelva a intentar la operación de importación de la pila.

Recuento de fallos de instancias de pila para las operaciones de StackSets

El recuento de fallos de las instancias de la pila notifica si las instancias de las pilas no se aprovisionan o actualizan. Estas instancias de pila no se implementaron por uno o más de los siguientes motivos:

- Recursos existentes con una configuración similar
- Falta de dependencias, como los roles de AWS Identity and Access Management (IAM)
- Otros factores en conflicto

Si desea implementar con la máxima simultaneidad, el recuento máximo de simultaneidad es uno más que el recuento de tolerancia a fallos, como máximo. Por ejemplo, si el recuento de tolerancia a fallos es 9, el recuento máximo de simultaneidad no puede ser superior a 10. Esto hará que la operación devuelva SUCCEEDED, incluso si algunas instancias de la pila no se actualizan. El nuevo recuento de fallos de instancias de pila le permite determinar si la operación solo se realizó correctamente de forma condicional, ya que el recuento de tolerancia a fallos está configurado para permitir todos los fallos.

Puede utilizar la Consola de administración de AWS, el AWS SDK o la AWS CLI para obtener el recuento de fallos y filtrar las instancias de la pila para determinar qué instancias deben reimplementarse.

Uso de la consola

Para visualizar la cantidad de instancias de pila fallidas:

1. Abra la [consola de CloudFormation](#) y seleccione StackSets.
2. Elija su StackSet y luego elija la pestaña Operations (Operaciones).
3. Elija un estado en la columna Status (Estado) para visualizar los detalles del estado. Encontrará la cantidad de instancias de pila fallidas para una operación particular en los detalles del estado.

Para ver la cuenta, la región y el estado de las instancias de pila de la operación:

1. En los detalles del estado, elija el recuento de instancias de pila fallidas. Ejemplo: Instancias de pila: **<number of failed stack instances>**.
2. Amplíe el panel lateral mediante la selección del encabezado del panel. Los resultados del panel lateral son los estados de las instancias de la pila después de completar la operación seleccionada.

Para ver los detalles de la instancia de pila actual de una operación:

1. Elija la pestaña Stack Instances (Instancias de la pila).
2. Filtre por Last operation ID (ID de última operación). Los resultados son los estados actuales y los motivos de estado de la última operación que modificó la instancia. Puede usar este filtro junto con Cuenta de AWS, Región de AWS, Estado detallado y Estado de desviación para refinar aún más los resultados de búsqueda.

Uso de AWS CLI

Para obtener la cantidad de instancias de pila fallidas, haga una llamada a `describe-stack-set-operation` o `list-stack-set-operations` y consulte `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
      }
    }
  ]
}

```

Para obtener una descripción general del historial de una operación en particular, utilice `list-stack-set-operation-results` para ver el estado y el motivo del estado de cada instancia de

pila cuando la operación esté completada. Consulte el siguiente ejemplo para ver el Status y el 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": ""
    }
]
}

```

Utilice `list-stack-instances` con los filtros `DETAILED_STATUS` y `LAST_OPERATION_ID` para obtener una lista de las instancias de pila que fallaron en la última operación que intentó implementar la instancia de pila. Observe la marca `--filters` en el ejemplo con `DETAILED_STATUS` y `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"
    }
]
}

```

A fin de encontrar el ID de la última operación para modificar una instancia de pila, use `list-stack-instances` o `describe-stack-instance` para obtener `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"
  }
}
```

Sincronización de pilas con código fuente almacenado en un repositorio de Git con sincronización de Git

Con la sincronización de Git, puede administrar sus pilas de CloudFormation con control de código fuente. Para ello, debe configurar CloudFormation para que monitoree un repositorio de Git. El repositorio se monitorea para detectar cambios en dos archivos:

- Un archivo de plantilla de CloudFormation que define una pila.
- Un archivo de implementación de pila que contiene los parámetros que configuran la pila.

Con la sincronización de Git, puede utilizar las solicitudes de extracción y el seguimiento de versiones para configurar, implementar y actualizar sus pilas de CloudFormation desde una ubicación centralizada. Al realizar cambios en la plantilla o el archivo de implementación, CloudFormation actualiza automáticamente la pila. Si utiliza solicitudes de extracción, CloudFormation puede dejar un comentario en la solicitud de extracción en la que se expliquen los cambios que se realizarán en la pila antes de actualizarla. Sin embargo, debe habilitar esta característica primero.

La sincronización de Git proporciona una interfaz de consola que puede usar para vincular a un repositorio, generar un archivo de implementación de la pila, actualizar una plantilla de CloudFormation y enviar una solicitud de extracción a su repositorio. La sincronización de Git también proporciona un panel de estado que puede usar para monitorear, editar y solucionar problemas de las implementaciones activas de las pilas de sincronización de Git. Se accede a la sincronización de Git a través de la [consola de CloudFormation](#) al [crear una pila](#). También, puede acceder a la sincronización de Git con CodeConnections. Para obtener más información, consulte [Cómo trabajar con configuraciones de sincronización para repositorios enlazados](#) en la Guía del usuario para desarrolladores sobre herramientas de la consola.

La sincronización de Git es compatible con los repositorios de [GitHub](#), [GitHub Enterprise](#), [GitLab](#), [Bitbucket](#) y [GitLab autoadministrado](#)

Note

La sincronización de Git está disponible en las siguientes regiones: EE. UU. Este (Norte de Virginia), EE. UU. Este (Ohio), EE. UU. Oeste (Norte de California), EE. UU. Oeste (Oregón), Canadá (Central), Asia Pacífico (Mumbai), Asia Pacífico (Tokio), Asia Pacífico (Seúl), Asia

Pacífico (Singapur), Asia Pacífico (Sídney), Europa (Irlanda), Europa (Londres), Europa (París), Europa (Estocolmo), Europa (Fráncfort), Europa (Milán) y América del Sur (São Paulo).

Para obtener información sobre el uso de la sincronización de Git con una estrategia de cuentas múltiples, consulte la siguiente entrada del blog [Usar la sincronización de CloudFormation Git para configurar los recursos en las cuentas de los clientes](#).

Temas

- [Cómo funciona la sincronización de Git con CloudFormation](#)
- [Requisitos previos para sincronizar pilas con un repositorio de Git mediante la sincronización de Git](#)
- [Creación de una pila a partir del código fuente del repositorio con sincronización de Git](#)
- [Cómo habilitar a CloudFormation para que publique un resumen de los cambios de pila en las solicitudes de cambios](#)
- [Panel de estado de sincronización de Git](#)

Cómo funciona la sincronización de Git con CloudFormation

En este tema se describe cómo funciona la sincronización de Git y se introducen los conceptos clave necesarios para trabajar con ella.

Temas

- [Cómo funciona la sincronización de Git](#)
- [Comentarios sobre solicitudes de extracción](#)
- [Archivo de implementación de la pila](#)
- [Archivo de plantilla de CloudFormation](#)
- [Repositorio de definiciones de plantilla](#)

Cómo funciona la sincronización de Git

Para utilizar la sincronización de Git, primero debe conectar un proveedor de Git a CloudFormation mediante el servicio [CodeConnections](#). En los procedimientos de esta guía, la conexión se crea a

través de la consola de CodeConnections. También puede crear la conexión con la AWS CLI. Puede utilizar cualquiera de los siguientes proveedores de Git:

- [GitHub](#)
- [GitHub Enterprise](#)
- [GitLab](#)
- [Bitbucket](#)
- [GitLab autoadministrado](#)

A continuación, cree una plantilla de CloudFormation que defina su pila y agréguela a su repositorio. Este archivo de plantilla está supervisado. CloudFormation actualiza la pila automáticamente cuando se realizan cambios en ella.

En la consola de CloudFormation, cree una nueva pila y elija Sincronizar desde Git para indicar a CloudFormation que utilice la sincronización de Git. Deberá especificar el repositorio y la rama que desea que CloudFormation supervise, y la plantilla de CloudFormation del repositorio que define la pila.

Durante la configuración, puede proporcionar su propio archivo de implementación de pila desde su repositorio o hacer que la sincronización de Git genere uno por usted. El archivo de implementación de la pila contiene parámetros y valores que configuran los recursos de la pila. Este archivo de implementación de pila está supervisado. CloudFormation actualiza la pila automáticamente cuando se realizan cambios en ella.

La sincronización de Git crea una solicitud de extracción en su repositorio para sincronizar su pila con el archivo de plantilla de CloudFormation y el archivo de implementación de la pila. Si la sincronización de Git genera el archivo de implementación de la pila por usted, la sincronización de Git lo envía a su repositorio.

A continuación, combine la solicitud de extracción con su repositorio para que CloudFormation aprovisiona la pila, la configure con sus parámetros de implementación y comience a monitorear su repositorio para detectar cambios.

A partir de ese momento, cada vez que realice cambios en el archivo de plantilla o en el archivo de implementación de la pila y los confirme en el repositorio, CloudFormation detectará automáticamente los cambios. Si su equipo usa solicitudes de extracción, los miembros de su equipo pueden revisar y aprobar los cambios antes de implementarlos. Una vez que se acepta la solicitud de cambios, CloudFormation los implementa.

Puede supervisar el estado de su configuración de sincronización de Git para la pila y ver un historial de las confirmaciones aplicadas a la pila en la consola de CloudFormation. La consola también proporciona herramientas para reconfigurar la sincronización de Git y solucionar problemas.

Comentarios sobre solicitudes de extracción

Puede elegir que CloudFormation cree un resumen de los cambios de código en la solicitud de extracción a través del servicio CodeConnections al activar la opción Habilitar comentarios en la solicitud de extracción en la consola. Al brindar un resumen de los cambios en las solicitudes de extracción, los miembros del equipo pueden revisar y comprender fácilmente el impacto de las modificaciones propuestas antes de fusionar la solicitud de extracción. Para obtener más información, consulte [Cómo habilitar a CloudFormation para que publique un resumen de los cambios de pila en las solicitudes de cambios](#).

Archivo de implementación de la pila

Un archivo de implementación de la pila es un archivo con notación de objetos JavaScript (JSON) o YAML que contiene parámetros y valores que administran la pila de CloudFormation. Este se supervisa para detectar cambios. Cuando los cambios en el archivo se confirman en el repositorio, la pila asociada se actualiza automáticamente.

El archivo de implementación de la pila contiene un par valor-clave y dos diccionarios:

- `template-file-path`

Esta es la ruta completa del repositorio del archivo de plantilla de CloudFormation. El archivo de plantilla declara los recursos de la pila de CloudFormation asociada a este archivo de implementación.

- `parameters`

El diccionario de parámetros contiene pares de clave-valor que configuran los recursos de la pila. Un archivo de implementación de la pila puede tener hasta 50 parámetros.

- `tags`

El diccionario de etiqueta contiene pares clave-valor que puede utilizar para identificar y clasificar los recursos de la pila. Un archivo de implementación de la pila puede tener hasta 50 etiquetas.

Puede proporcionar su propio archivo de implementación de la pila o hacer que la sincronización de Git cree uno por usted y envíe automáticamente una solicitud de extracción a su repositorio.

Puede gestionar los parámetros y las etiquetas editando el archivo de implementación de la pila y confirmando los cambios en el repositorio.

A continuación, se muestra un ejemplo de un archivo de implementación de pila de sincronización de 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'
```

Archivo de plantilla de CloudFormation

Un archivo de plantilla contiene una declaración de los recursos de AWS que componen una pila de CloudFormation. Con la sincronización de Git, el archivo de plantilla se almacena en su repositorio de Git y el archivo de implementación de la pila hace referencia a él. Puede gestionar la pila editando el archivo de plantilla y confirmando los cambios en el repositorio.

Para obtener más información, consulte [Trabajo con plantillas de CloudFormation](#).

Repositorio de definiciones de plantilla

El repositorio de definiciones de plantillas es el repositorio de Git que está vinculado a CloudFormation mediante la sincronización de Git. El repositorio se supervisa para detectar cambios en la plantilla de CloudFormation y en el archivo de implementación de la pila. Cuando los cambios en el archivo se confirman en el repositorio, la pila asociada se actualiza automáticamente.

Important

Al configurar el repositorio de definiciones de plantillas en la consola de sincronización de Git, seleccione el repositorio y la rama correctos en la conexión de Git. La sincronización de Git solo supervisa la rama y el repositorio configurados para detectar cambios en la plantilla de CloudFormation y los archivos de implementación de la pila.

Requisitos previos para sincronizar pilas con un repositorio de Git mediante la sincronización de Git

Antes de sincronizar una pila de CloudFormation con su repositorio de Git, compruebe que se cumplan los siguientes requisitos.

Temas

- [Repositorio de Git](#)
- [Plantilla de CloudFormation](#)
- [Rol de servicio de sincronización de Git](#)
- [Permisos de IAM para los usuarios de la consola](#)

Repositorio de Git

Debe tener un repositorio de Git alojado en una de las siguientes plataformas.

- [GitHub](#)
- [GitHub Enterprise](#)
- [GitLab](#)
- [Bitbucket](#)
- [GitLab autoadministrado](#)

El repositorio puede ser público o privado. Deberá conectar este repositorio de Git a CloudFormation a través de la [consola de Conexiones](#).

Plantilla de CloudFormation

El repositorio de Git debe contener un archivo de [plantilla de CloudFormation](#) verificado en la ramificación que pretende conectar con la sincronización de Git. El [archivo de implementación de la pila](#) hará referencia a esta plantilla.

Rol de servicio de sincronización de Git

La sincronización de Git requiere un rol de IAM. Puede elegir que se cree un rol de IAM para la pila al configurar la sincronización de Git o puede utilizar un rol existente.

 Note

Un rol de IAM generado automáticamente solo aplica permisos a la pila para la que se generó el rol. Para volver a utilizar un rol de IAM generado automáticamente, debe editar el rol de la nueva pila.

Permisos necesarios para la creación de un rol de servicio de sincronización de Git

El rol de IAM que proporcione para la sincronización de Git requiere los siguientes permisos.

- `cloudformation:CreateChangeSet`
- `cloudformation>DeleteChangeSet`
- `cloudformation:DescribeChangeSet`
- `cloudformation:DescribeStackEvents`
- `cloudformation:DescribeStacks`
- `cloudformation:ExecuteChangeSet`
- `cloudformation:ListChangeSets`
- `cloudformation:ValidateTemplate`
- `events:PutRule`
- `events:PutTargets`

 Note

Los permisos necesarios anteriores se agregan automáticamente a los roles de IAM que genera la sincronización de Git.

El siguiente ejemplo de rol de IAM incluye los permisos para la sincronización de Git.

JSON

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

```

        "Sid": "SyncToCloudFormation",
        "Effect": "Allow",
        "Action": [
            "cloudformation:CreateChangeSet",
            "cloudformation:DeleteChangeSet",
            "cloudformation:DescribeChangeSet",
            "cloudformation:DescribeStackEvents",
            "cloudformation:DescribeStacks",
            "cloudformation:ExecuteChangeSet",
            "cloudformation:GetTemplate",
            "cloudformation:ListChangeSets",
            "cloudformation:ListStacks",
            "cloudformation:ValidateTemplate"
        ],
        "Resource": "*"
    },
    {
        "Sid": "PolicyForManagedRules",
        "Effect": "Allow",
        "Action": [
            "events:PutRule",
            "events:PutTargets"
        ],
        "Resource": "*",
        "Condition": {
            "StringEquals": {
                "events:ManagedBy":
["cloudformation.sync.codeconnections.amazonaws.com"]
            }
        }
    },
    {
        "Sid": "PolicyForDescribingRule",
        "Effect": "Allow",
        "Action": "events:DescribeRule",
        "Resource": "*"
    }
]
}

```

Política de confianza

Debe proporcionar la siguiente política de confianza a la hora de crear el rol para definir la relación de confianza.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CfnGitSyncTrustPolicy",
      "Effect": "Allow",
      "Principal": {
        "Service": "cloudformation.sync.codeconnections.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Le recomendamos que utilice las claves de condición `aws:SourceArn` y `aws:SourceAccount` para protegerse contra el problema del suplente confuso. La cuenta de origen es el ID de su cuenta y el ARN de origen es el ARN de la conexión en el servicio de [CodeConnections](#) que permite a CloudFormation conectarse a su repositorio de Git.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CfnGitSyncTrustPolicy",
      "Effect": "Allow",
      "Principal": {
        "Service": "cloudformation.sync.codeconnections.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {

```

```
    "aws:SourceAccount": "123456789012"
  },
  "ArnLike": {
    "aws:SourceArn": "arn:aws:codeconnections:us-
east-1:123456789012:connection/EXAMPLE64-8aad-4d5d-8878-dfcab0bc441f"
  }
}
]
```

Para obtener más información sobre el problema del suplente confuso, consulte [Prevención de la sustitución confusa entre servicios](#).

Permisos de IAM para los usuarios de la consola

Para configurar correctamente la sincronización de Git a través de la consola CloudFormation, los usuarios finales también deben tener permisos a través de IAM.

Se requieren los siguientes permisos de codeconnections para crear y administrar la conexión a su repositorio Git.

- codeconnections:CreateRepositoryLink
- codeconnections:CreateSyncConfiguration
- codeconnections>DeleteRepositoryLink
- codeconnections>DeleteSyncConfiguration
- codeconnections:GetRepositoryLink
- codeconnections:GetSyncConfiguration
- codeconnections:ListRepositoryLinks
- codeconnections:ListSyncConfigurations
- codeconnections:ListTagsForResource
- codeconnections:TagResource
- codeconnections:UntagResource
- codeconnections:UpdateRepositoryLink
- codeconnections:UpdateSyncBlocker
- codeconnections:UpdateSyncConfiguration

- `codeconnections:UseConnection`

Los usuarios de la consola también deben tener los siguientes permisos de `cloudformation` para ver y administrar las pilas durante el proceso de configuración de la sincronización de Git.

- `cloudformation:CreateChangeSet`
- `cloudformation>DeleteChangeSet`
- `cloudformation:DescribeChangeSet`
- `cloudformation:DescribeStackEvents`
- `cloudformation:DescribeStacks`
- `cloudformation:ExecuteChangeSet`
- `cloudformation:GetTemplate`
- `cloudformation:ListChangeSets`
- `cloudformation:ListStacks`
- `cloudformation:ValidateTemplate`

Note

Aunque los permisos del conjunto de cambios (`cloudformation:CreateChangeSet`, `cloudformation>DeleteChangeSet`, `cloudformation:DescribeChangeSet`, `cloudformation:ExecuteChangeSet`) podrían no ser estrictamente necesarios para el uso exclusivo de la consola, se recomiendan para habilitar las capacidades de inspección y gestión del full stack.

El siguiente ejemplo de política de IAM incluye los permisos de usuario necesarios para configurar la sincronización de Git a través de la consola.

JSON

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

```

    "Effect": "Allow",
    "Action": [
        "codeconnections:CreateRepositoryLink",
        "codeconnections:CreateSyncConfiguration",
        "codeconnections>DeleteRepositoryLink",
        "codeconnections>DeleteSyncConfiguration",
        "codeconnections:GetRepositoryLink",
        "codeconnections:GetSyncConfiguration",
        "codeconnections:ListRepositoryLinks",
        "codeconnections:ListSyncConfigurations",
        "codeconnections:ListTagsForResource",
        "codeconnections:TagResource",
        "codeconnections:UntagResource",
        "codeconnections:UpdateRepositoryLink",
        "codeconnections:UpdateSyncBlocker",
        "codeconnections:UpdateSyncConfiguration",
        "codeconnections:UseConnection",
        "codeconnections:CreateForcedTargetSync",
        "codeconnections:CreatePullRequestForResource"
    ],
    "Resource": "*"
},
{
    "Sid": "CloudFormationConsolePermissions",
    "Effect": "Allow",
    "Action": [
        "cloudformation:CreateChangeSet",
        "cloudformation>DeleteChangeSet",
        "cloudformation:DescribeChangeSet",
        "cloudformation:DescribeStackEvents",
        "cloudformation:DescribeStacks",
        "cloudformation:ExecuteChangeSet",
        "cloudformation:GetTemplate",
        "cloudformation:ListChangeSets",
        "cloudformation:ListStacks",
        "cloudformation:ValidateTemplate"
    ],
    "Resource": "*"
}
]
}

```

 Note

Al crear una política de IAM que incluya los permisos `codeconnections:CreateForcedTargetSync` y `codeconnections:CreatePullRequestForResource`, es posible que usted vea una advertencia en la consola de IAM que indique que estas acciones no existen. Esta advertencia se puede ignorar y la política se seguirá creando correctamente. Estos permisos son necesarios para determinadas operaciones de sincronización de Git a pesar de que la consola de IAM no los reconozca.

Creación de una pila a partir del código fuente del repositorio con sincronización de Git

En este tema se explica cómo crear una pila de CloudFormation que se sincronice con un repositorio de Git mediante la sincronización de Git.

 Important

Antes de continuar, complete todos los [requisitos previos](#) de la sección anterior.

Creación de una pila a partir del código fuente del repositorio

1. Inicie sesión en la Consola de administración de AWS y abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation>.
2. En la barra de navegación de la parte superior de la pantalla, elija la Región de AWS en la que creó la pila.
3. En la página Pilas, elija Crear pila y, a continuación, elija Con recursos nuevos (estándar).
4. En la página Crear pila, proceda del modo siguiente:
 - a. En Requisito previo: preparar la plantilla, mantenga seleccionada la opción Seleccione una plantilla existente.
 - b. En Especificar plantilla, elija Sincronizar desde Git y, a continuación, seleccione Siguiente.
5. En la página Especificar los detalles de la pila, en Nombre de la pila, ingrese un nombre para la pila. Los nombres de pilas pueden incluir letras (A-Z y a-z), números (0-9) y guiones (-).

6. En Archivo de implementación de Stack, Creación del archivo de implementación:
 - Si no ha creado un archivo de implementación de pilas y lo ha agregado al repositorio, elija Crear el archivo con los siguientes parámetros y colocarlo en mi repositorio.
 - Si tiene archivo de implementación de pila en su repositorio, seleccione Proporciono mi propio archivo en mi repositorio.
7. En Repositorio de definiciones de plantillas, seleccione Elegir un repositorio de Git vinculado para seleccionar un repositorio de Git que ya está vinculado a CloudFormation o Vincular un repositorio de Git para vincular uno nuevo. Si elige Vincular un repositorio de Git, haga lo siguiente:
 - a. En Seleccionar el proveedor de repositorios, elija una de las siguientes opciones:
 - GitHub
 - GitHub Enterprise Server
 - GitLab
 - Bitbucket
 - GitLab autoadministrado
 - b. Para Conexión, elija una conexión de la lista. Si no aparece ninguna opción en la lista Conexión, elija agregar una nueva conexión para ir a la [Consola de conexiones](#) y crear una conexión con su repositorio.
8. En la lista de Repositorios, seleccione el repositorio de Git que contenga el archivo de plantillas de la pila.
9. En la lista de Ramas, seleccione la rama que quiere que supervise la sincronización de Git sync.

 Note

La sincronización de Git solo supervisa la rama seleccionada para detectar cambios en la plantilla de CloudFormation y los archivos de implementación de la pila. Cualquier cambio que desee aplicar a su pila debe estar registrado en esta rama.

10. Para la Ruta del archivo de implementación, especifique la ruta completa, incluido el nombre del archivo de implementación de la pila, desde la raíz de la rama de repositorio.

Si CloudFormation genera el archivo por usted, aquí es donde se guardará el archivo en su repositorio. Si proporciona el archivo, esta es la ubicación del archivo en su repositorio.

11. Agregar un rol de IAM. El rol de IAM incluye los permisos necesarios para que CloudFormation sincronice la pila desde su repositorio de Git. Puede elegir Nuevo rol de IAM para generar un nuevo rol o elegir Rol de IAM existente para seleccionar un rol existente de su Cuenta de AWS. Si decide generar un nuevo rol, los permisos necesarios se incluyen en el rol.
12. Cómo habilitar o desactivar los comentarios en una solicitud de extracción:
 - Para incluir la información posterior al conjunto de cambios de CloudFormation en las solicitudes de cambios para actualizaciones de pila, mantenga activada la opción Habilitar comentarios en las solicitudes de cambios.
 - Si desactiva esta opción, CloudFormation no describirá las diferencias entre la configuración de la pila actual y los cambios propuestos en las solicitudes de incorporación de cambios cuando se actualicen los archivos del repositorio.
13. Para Ruta del archivo de plantillas, especifique la ruta completa desde la raíz de su repositorio para el archivo de plantillas de la pila.
14. (Opcional) Para especificar los parámetros de la pila, seleccione Agregar parámetro, brinde una clave y un valor para cada parámetro y, a continuación, seleccione Siguiente. Para obtener más información, consulte [Archivo de implementación de la pila](#).

Por ejemplo, para especificar un parámetro **port=8080** en el archivo de implementación de la pila, haga lo siguiente:

- a. Elija Add parameter (Añadir parámetro).
 - b. En Clave, escriba **port**.
 - c. En Valor, introduzca **8080**.
15. (Opcional) Para especificar etiquetas de pila, seleccione Agregar nueva etiqueta, ingrese un valor y una clave de etiqueta, y luego elija Siguiente. Para obtener más información, consulte [Archivo de implementación de la pila](#).
 16. Seleccione Siguiente para continuar con Configurar las opciones de la pila. Para obtener más información acerca de cómo configurar opciones de la pila, consulte [Configurar las opciones de la pila](#).

Cuando haya completado la configuración de la pila, seleccione Siguiente para continuar.

17. Revise la configuración de la pila y confirme lo siguiente:
 - La plantilla de la pila se configuró correctamente y se estableció en Sincronizar desde Git.
 - El archivo de implementación se configuró correctamente.

- El repositorio de definiciones de plantillas se configuró correctamente y, en particular, se seleccionó el Repositorio y Nombre de rama correctos.
- La vista previa del archivo de implementación es correcta y contiene los parámetros y valores esperados.

18. Seleccione Enviar para crear la pila.

Después de que selecciona Enviar, se crea automáticamente una solicitud de extracción en su repositorio de Git. Debe combinar esta solicitud de extracción en su repositorio de Git para crear su pila. Una vez que se crea la pila, CloudFormation supervisa su repositorio de Git en busca de cambios.

Actualización de la pila desde el repositorio de Git

Para actualizar la pila, realice los cambios directamente en el archivo de plantilla o en el archivo de implementación de la pila en su repositorio de Git. Tras confirmar los cambios en la rama supervisada, CloudFormation actualiza automáticamente la pila. Si utiliza solicitudes de extracción, se crea automáticamente una solicitud de extracción en su repositorio de Git antes de que se actualice la pila. Debe combinar esta solicitud de extracción en su repositorio de Git para actualizar su pila.

En la consola de CloudFormation, puede seleccionar la pila y elegir la pestaña Sincronización de Git para ver información sobre el estado de la pila y los eventos de sincronización. Para obtener más información, consulte [Panel de estado de sincronización de Git](#).

Cómo habilitar a CloudFormation para que publique un resumen de los cambios de pila en las solicitudes de cambios

En este tema, se muestra cómo habilitar CloudFormation para publicar un resumen de los cambios de pila en las solicitudes de incorporación de cambios en tu repositorio de Git.

Al habilitar la característica de comentarios en las solicitudes de incorporación de cambios, permite que CloudFormation publique un comentario que describa las diferencias entre la configuración de la pila actual y los cambios propuestos cuando se actualicen los archivos del repositorio. Este comentario ofrece un resumen de los recursos que se agregarán, modificarán o eliminarán, lo que permitirá realizar una revisión exhaustiva del código antes de fusionar la solicitud de extracción.

Habilitación de los comentarios en las solicitudes de incorporación de cambios para una nueva pila (consola)

Al crear la pila, en la página Especificar los detalles de la pila, en el Repositorio de definiciones de plantillas, asegúrese de que la opción Habilitar comentarios en las solicitudes de extracción esté activada. Esta es la configuración predeterminada para las nuevas pilas.

Habilitación de los comentarios en las solicitudes de extracción de una nueva pila (consola)

1. Inicie sesión en la Consola de administración de AWS y abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation>.
2. En la barra de navegación de la parte superior de la pantalla, seleccione la Región de AWS en la que creó su pila.
3. En la página Pilas, seleccione la pila en ejecución que desea actualizar.
4. Elija la pestaña Sincronización con Git y, a continuación, Editar.
5. En la página Editar configuración de sincronización con Git, en Repositorio de definiciones de plantillas, active la opción Habilitar comentarios en solicitudes de extracción.
6. Seleccione Update configuration (Actualizar la configuración).

Panel de estado de sincronización de Git

Para ver el estado de una implementación de sincronización de Git AWS CloudFormation, seleccione la pila en la consola de CloudFormation y elija la pestaña de Sincronización de Git.

La pestaña de Sincronización de Git se divide en dos paneles: Estado de sincronización de Git y Últimos eventos de sincronización.

Estado de sincronización de Git

El panel superior proporciona la siguiente información sobre la configuración de sincronización de Git para la pila.

Repositorio

Un enlace al repositorio que está conectado a la sincronización de Git

Proveedor de repositorio

El nombre del proveedor de repositorio

Rama

El nombre de la rama que supervisa la sincronización de Git

Ruta del archivo de implementación

La ruta completa al archivo de implementación de la pila para esta pila

Estado de sincronización del repositorio

El estado de la operación de sincronización más reciente

Mensaje de estado de sincronización del repositorio

El mensaje de la operación de sincronización más reciente

Estado de sincronización de Git

El estado de la sincronización de Git para esta pila

Estado de aprovisionamiento

El estado de la operación de aprovisionamiento

En la parte superior derecha del panel, use los siguientes botones para modificar o actualizar la sincronización de Git:

- Editar: edita la configuración de sincronización de Git.
- Reintentar la última confirmación: actualiza la pila de acuerdo con la confirmación más reciente en el repositorio.
- Desconectar: desconecta la sincronización de Git de la pila.
- Actualizar: actualiza el panel de estado de sincronización de Git.

Últimos eventos de sincronización

El panel de Eventos de sincronización más recientes muestra una tabla de confirmaciones que se aplicaron a la pila.

Puede ordenar la tabla con las flechas del encabezado de cada columna. La tabla se puede ordenar de forma ascendente o descendente de acuerdo con lo siguiente:

- Fecha
- ID de confirmación

- Evento
- Fecha
- Tipo de evento

Estados de pilas admitidos

La sincronización de Git solo se puede configurar para una pila si la pila se encuentra en uno de los siguientes estados admitidos:

- CREATE_COMPLETE
- UPDATE_COMPLETE
- UPDATE_ROLLBACK_COMPLETE
- IMPORT_COMPLETE
- IMPORT_ROLLBACK_COMPLETE

La siguiente tabla contiene una lista completa de los códigos de estado de la pila con descripciones:

Estado de la pila y estado detallado opcional	Descripción
CREATE_COMPLETE	Creación correcta de una o varias pilas.
CREATE_IN_PROGRESS	Creación continua de una o varias pilas.
CREATE_FAILED	Creación incorrecta de una o varias pilas. Consulte los eventos de pila para ver los mensajes de error asociados . Algunos de los posibles motivos de que la creación no se haya realizado correctamente incluyen la falta de permisos para trabajar con todos los recursos de pila, el rechazo de los valores de los parámetros por parte de un servicio de AWS o que se ha agotado el tiempo de espera durante la creación del recurso.
DELETE_COMPLETE	Eliminación correcta de una o varias pilas. Las pilas eliminadas se conservan y se pueden ver durante 90 días.

Estado de la pila y estado detallado opcional	Descripción
DELETE_FAILED	Eliminación incorrecta de una o varias pilas. Dado que la eliminación no se ha realizado correctamente, es posible que algunos recursos aún se sigan ejecutando; sin embargo, no puede trabajar ni actualizar la pila. Vuelva a eliminar la pila o consulte los eventos de la pila para ver los mensajes de error asociados.
DELETE_IN_PROGRESS	Eliminación continua de una o varias pilas.
REVIEW_IN_PROGRESS	Creación continua de una o varias pilas con un StackId esperado, pero sin plantillas ni recursos.  Important Una pila con este código de estado cuenta para el número máximo posible de pilas.
ROLLBACK_COMPLETE	Eliminación correcta de una o varias pilas después de un error en la creación de la pila o después de que se haya cancelado explícitamente la creación de la pila. La pila vuelve al estado anterior de trabajo. Se eliminan todos los recursos que se hayan creado durante la acción de creación de pila. Este estado solo existe después de la creación fallida de una pila. Esto significa que todas las operaciones de la pila creada parcialmente se ha limpiado de forma apropiada. Cuando en este estado, solo puede realizarse una operación de eliminación.
ROLLBACK_FAILED	Eliminación incorrecta de una o varias pilas después de un error en la creación de pila o después de que se haya cancelado explícitamente la creación de pila. Elimine la pila o consulte los eventos de pila para ver los mensajes de error asociados.

Estado de la pila y estado detallado opcional	Descripción
ROLLBACK_IN_PROGRESS	Eliminación continua de una o varias pilas después de un error en la creación de pila o después de que se haya cancelado explícitamente la creación de pila.
UPDATE_COMPLETE	Actualización correcta de una o varias pilas.
UPDATE_COMPLETE_CLEANUP_IN_PROGRESS	Eliminación continua de recursos antiguos para una o varias pilas después de una actualización correcta de las pilas. Para las actualizaciones de pilas que requieren que se sustituyan los recursos, CloudFormation crea los nuevos recursos en primer lugar y, a continuación, elimina los antiguos para ayudar a reducir interrupciones con la pila. En este estado, la pila se ha actualizado y se puede utilizar, pero CloudFormation sigue eliminando los recursos antiguos.
UPDATE_FAILED	Actualización incorrecta de una o varias pilas. Consulte los eventos de pila para ver los mensajes de error asociados.
UPDATE_IN_PROGRESS	Actualización continua de una o varias pilas.
UPDATE_ROLLBACK_COMPLETE	Recuperación correcta del estado activo anterior de una o varias pilas después de un error en la actualización de las pilas.
UPDATE_ROLLBACK_COMPLETE_CLEANUP_IN_PROGRESS	Eliminación continua de recursos nuevos para una o varias pilas después de una actualización incorrecta de las pilas. En este estado, se ha restaurado el estado anterior de la pila y se puede utilizar, pero CloudFormation sigue eliminando los nuevos recursos que ha creado durante la actualización de la pila.

Estado de la pila y estado detallado opcional	Descripción
UPDATE_ROLLBACK_FAILED	Restauración incorrecta del estado activo anterior de una o varias pilas después de un error en la actualización de las pilas. En este estado, puede eliminar la pila o continuar con la restauración . Probablemente necesite corregir errores antes de que la pila vuelva a estar operativa. También puede contactar con Soporte para restaurar la pila a un estado útil.
UPDATE_ROLLBACK_IN_PROGRESS	Recuperación continua del estado activo anterior de una o varias pilas después de un error en la actualización de las pilas.
IMPORT_IN_PROGRESS	La operación de importación está actualmente en curso.
IMPORT_COMPLETE	La operación de importación se ha completado correctamente para todos los recursos de la pila compatibles con <code>resource import</code> .
IMPORT_ROLLBACK_IN_PROGRESS	La importación volverá a la configuración de la plantilla anterior.
IMPORT_ROLLBACK_FAILED	La operación de restauración de importación ha generado un error para un recurso de la pila como mínimo. Los resultados estarán disponibles para los recursos que CloudFormation haya importado correctamente.
IMPORT_ROLLBACK_COMPLETE	La importación ha vuelto correctamente a la configuración de plantilla anterior.

Administración de extensiones a través del registro de CloudFormation

El registro de CloudFormation sirve como centro neurálgico para administrar las extensiones que pueden integrarse en las plantillas de CloudFormation en su Cuenta de AWS. Las extensiones incluyen tipos de recursos, módulos y enlaces de AWS y publicadores de terceros, así como sus propias extensiones personalizadas. El registro facilita la detección y el aprovisionamiento de extensiones en sus plantillas de CloudFormation de la misma manera que usa recursos aprovisionados por AWS.

En esta sección se describe el uso del registro de CloudFormation para activar las extensiones de terceros en su cuenta, incluidas las siguientes:

- Activación de extensiones públicas
- Registro y activación de extensiones privadas

Temas

- [Documentación relacionada](#)
- [Conceptos de registro de CloudFormation](#)
- [Visualización de las extensiones disponibles y activadas en el registro de CloudFormation](#)
- [Utilización de extensiones públicas de terceros del registro de CloudFormation](#)
- [Uso de extensiones privadas de terceros que se han compartido con usted](#)
- [Edición de los datos de configuración de las extensiones en su cuenta](#)
- [Registro de tipos de recursos en AWS Config](#)

Documentación relacionada

Si es un desarrollador interesado en crear sus propias extensiones, consulte la siguiente documentación:

- [Desarrollo de módulos mediante la CLI de CloudFormation](#) en la Guía del usuario de la interfaz de línea de comandos de CloudFormation
- [Creación de tipos de recursos mediante la CLI de CloudFormation](#) en la Guía del usuario de la interfaz de línea de comandos de CloudFormation

- [Desarrollo de enlaces personalizados mediante la CLI de CloudFormation](#) en la Guía del usuario de enlaces de CloudFormation

Además, todos los tipos de recursos de AWS aprovisionables disponibles en el registro de CloudFormation se pueden usar con el API de control de nube de AWS, con sus atributos y propiedades definidos en un esquema JSON estándar. Para obtener más información, consulte la [Guía del usuario de la API de Cloud Control](#). Cuando utilice la API de Cloud Control para realizar operaciones CRUDL (Crear, Leer, Actualizar, Eliminar, Enumerar) en recursos de AWS, solo podrá hacerlo en los recursos de AWS dentro de su propia Cuenta de AWS.

Conceptos de registro de CloudFormation

En este tema, se explican los conceptos clave que lo ayudarán a aprender acerca del registro de CloudFormation y a empezar a utilizarlo.

Tipos de extensiones

El registro de CloudFormation ofrece los siguientes tipos de extensiones:

Enlaces

Los enlaces son comprobaciones de validación que inspeccionan las pilas o los recursos específicos antes de que CloudFormation los cree, actualice o elimine. Además, los enlaces se pueden invocar durante una operación de creación de conjuntos de cambios. Proporcionan un mecanismo para hacer cumplir los estándares y las mejores prácticas de la organización al validar las configuraciones de los recursos en función de requisitos específicos. Si un enlace detecta alguna configuración que no cumple con su lógica, puede emitir una advertencia o fallar en el proceso de aprovisionamiento para evitar que se implementen los recursos que no cumplen con los requisitos. Para obtener más información, consulte la [Guía del usuario de enlaces de CloudFormation](#).

Para obtener documentación sobre cómo configurar Hooks mediante la consola de CloudFormation, consulte las siguientes secciones de la Guía del usuario de Hooks CloudFormation:

- [controles proactivos AWS Control Tower como Hooks](#)
- [Enlaces de Guard](#)
- [Enlaces de Lambda](#)

Modules

Los módulos son configuraciones de recursos reutilizables que se pueden incluir en varias plantillas de pila de CloudFormation. Simplifican la creación y el mantenimiento de las plantillas de CloudFormation al encapsular configuraciones de recursos complejas o de uso frecuente en componentes reutilizables. Esto promueve la coherencia y la estandarización en todas las implementaciones de infraestructura de su organización.

Tipos de recurso

Los tipos de recursos le permiten modelar y automatizar recursos de terceros o recursos personalizados que CloudFormation no admite de forma nativa. Al desarrollar los tipos de recursos, puede ampliar las capacidades de CloudFormation para aprovisionar y administrar los recursos de varios servicios de terceros.

Extensiones públicas

Las extensiones públicas son extensiones de CloudFormation que se publican en el registro para que las usen todos los Cuentas de AWS. Entre ellos se incluyen:

- **Extensiones de AWS:** las extensiones publicadas por AWS son siempre públicas y están activadas por defecto, por lo que no tiene que realizar ninguna acción antes de utilizarlas en su cuenta. AWS es el control de versiones de estas extensiones, por lo que siempre está utilizando la última versión disponible.
- **Extensiones de terceros:** son extensiones que están disponibles para uso general por editores distintos de AWS. Para usar una extensión de terceros, antes debe activarla en su cuenta y región.

Extensiones activadas

El registro de CloudFormation en su Cuenta de AWS específica incluye tres tipos de extensiones activadas:

- **Extensiones de AWS:** todas las extensiones públicas de AWS se activan automáticamente.
- **Extensiones de terceros activadas:** son copias locales de extensiones de terceros públicas que ha activado explícitamente para su cuenta y región. Cuando activa una extensión pública de terceros, CloudFormation crea una copia local en el registro de su cuenta. Para obtener más información, consulte [Utilización de extensiones públicas de terceros del registro de CloudFormation](#).

- Registradas de forma privada: son extensiones privadas que no se muestran en el registro público de CloudFormation. Estas pueden ser extensiones que haya creado usted o las que se le hayan compartido a través de la organización u otros terceros. Para utilizar dicha extensión privada en la cuenta, primero debe registrarla. Al registrar la extensión, se carga una copia en el registro de CloudFormation de su cuenta y se activa. Para obtener más información, consulte [Uso de extensiones privadas de terceros que se han compartido con usted](#).

El uso de extensiones registradas de forma privada y extensiones públicas activadas de editores de terceros en su cuenta es como utilizarlas en un entorno de pruebas. Las extensiones se gestionan con control de versiones, por lo que su comportamiento de aprovisionamiento está vinculado a una versión específica. Como resultado, estas extensiones funcionan de la misma manera que las extensiones públicas y siguen las mismas reglas específicas de versión.

 Note

Las extensiones registradas de forma privada y las extensiones públicas de terceros activadas pueden implementar controladores de eventos que se ejecutan durante las operaciones de crear, leer, actualizar, listar y eliminar. El uso de estas extensiones en sus pilas de CloudFormation puede generar cargos en su cuenta, además de los cargos por los recursos creados. Para obtener más información, consulte [Precios de CloudFormation](#).

Visualización de las extensiones disponibles y activadas en el registro de CloudFormation

Para ver las extensiones disponibles y activadas en el registro de CloudFormation, puede usar la Consola de administración de AWS o la AWS CLI.

Visualización de las extensiones disponibles y activadas (consola)

1. Inicie sesión en la Consola de administración de AWS y abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation>.
2. En la barra de navegación de la parte superior de la pantalla, elija la Región de AWS.
3. Desde el panel de navegación, en Registro, seleccione la categoría de extensión que quiere ver:
 - Public extensions (Extensiones públicas) muestra las extensiones públicas disponibles en su cuenta.

1. En Filtrar, Tipo de extensión, elija su tipo de extensión: Tipos de recursos, Módulos o Enlaces.
2. En Filtrar, Publicador, elija un publicador: AWS o un Tercero.
- Activated extensions (Extensiones activadas) muestra las extensiones públicas y privadas activadas en su cuenta.
 1. Elija el tipo de extensión: Resource types (Tipos de recurso), Modules (Módulos) o Hooks (Enlaces).
 2. Use el menú desplegable Filtrar para seleccionar las extensiones que se van a ver:
 - AWS – muestra las extensiones publicadas por AWS. Las extensiones publicadas por AWS están activadas de forma predeterminada.
 - Third-party (Terceros): enumera las extensiones públicas de los editores distintos de AWS que ha activado en esta cuenta.
 - Registered (Registradas): muestra cualquier extensión privada que haya activado en esta cuenta.
 - Publisher (Editor): muestra cualquier extensión pública que haya publicado en esta cuenta.
4. Busque o seleccione el nombre de la extensión para ver los detalles de la extensión.

Visualización de las extensiones disponibles y activadas (AWS CLI)

Utilice el comando [list-types](#).

Utilización de extensiones públicas de terceros del registro de CloudFormation

Para utilizar una extensión pública de terceros en su plantilla, antes debe activar la extensión para la cuenta y región en la que desea utilizarla. Al activar una extensión, esta se podrá utilizar en operaciones de pila en la cuenta y la región en la que se active.

Cuando se activa una extensión pública de terceros, CloudFormation crea una entrada en el registro de extensiones de su cuenta para la extensión activada como una extensión privada. Esto le permite establecer cualquier propiedad de configuración que incluya la extensión. Las propiedades de configuración definen cómo se configura la extensión para una Cuenta de AWS y región determinadas.

Además de establecer las propiedades de configuración, también puede personalizar la extensión de las siguientes maneras:

- Especifique el rol de ejecución que CloudFormation utiliza para activar la extensión, además de configurar el registro para la extensión.
- Especifique si la extensión se actualiza automáticamente cuando esté disponible una nueva versión secundaria o de parche.
- Especifique un alias para usar en lugar del nombre de la extensión pública de terceros. Esto puede ayudar a evitar colisiones de nombres entre extensiones de terceros.

Temas

- [Configure un rol de ejecución con permisos de IAM y una política de confianza para el acceso a las extensiones públicas](#)
- [Utilice automáticamente nuevas versiones de extensiones](#)
- [Utilización de alias para hacer referencia a extensiones](#)
- [Comandos de la AWS CLI de uso común para trabajar con extensiones públicas](#)
- [Activación de una extensión pública de terceros en su cuenta](#)
- [Actualización de una extensión pública de terceros en su cuenta](#)
- [Desactivación de extensiones públicas de terceros en su cuenta](#)

Configure un rol de ejecución con permisos de IAM y una política de confianza para el acceso a las extensiones públicas

Al activar una extensión pública desde el registro de CloudFormation, puede proporcionar un rol de ejecución que otorgue a CloudFormation los permisos necesarios para invocar esa extensión en su Cuenta de AWS y en su región.

Los permisos necesarios para el rol de ejecución se definen en la sección del controlador del esquema de extensión. Debe crear una política de IAM que conceda los permisos específicos que necesita la extensión y adjuntarla al rol de ejecución.

Además de la política de permisos, el rol de ejecución también necesitará tener una política de confianza que permita a CloudFormation asumir el rol. Siga las instrucciones sobre cómo [crear un rol mediante políticas de confianza personalizadas](#) de la Guía del usuario de IAM para crear un rol con una política de confianza personalizada.

Relación de confianza

A continuación se muestra un ejemplo de políticas de confianza que puede utilizar.

De forma opcional, puede restringir el alcance del permiso para evitar que se produzcan errores confusos entre servicios mediante el uso de una o más claves de contexto de condiciones globales en el campo `Condition`. Para obtener más información, consulte [Prevención de la sustitución confusa entre servicios](#).

- Configure el valor `aws:SourceAccount` en el ID de su cuenta.
- Establezca el valor `aws:SourceArn` en el ARN de su extensión.

Política de confianza de ejemplo 1

A continuación, se muestra un ejemplo de política de confianza de rol de IAM para la extensión del tipo de recurso.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "resources.cloudformation.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:cloudformation:us-west-2:123456789012:type/resource/Organization-Service-Resource"
        }
      }
    }
  ]
}
```

Política de confianza de ejemplo 2

A continuación, se muestra un ejemplo de política de confianza de rol de IAM para una extensión de enlace.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "resources.cloudformation.amazonaws.com",
          "hooks.cloudformation.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:cloudformation:us-west-2:123456789012:type/hook/Organization-Service-Hook"
        }
      }
    }
  ]
}
```

Utilice automáticamente nuevas versiones de extensiones

Cuando se activa una extensión, también puede especificar el tipo de extensión para utilizar la versión secundaria más reciente. El tipo de extensión actualiza la versión secundaria siempre que el editor publique una nueva versión en la extensión activada.

Por ejemplo, la próxima vez que lleve a cabo una operación de pila, como crear o actualizar una pila, con una plantilla que incluya esa extensión, CloudFormation utilizará la nueva versión secundaria.

La actualización a una nueva versión de extensión, ya sea de forma automática o manual, no afecta a ninguna instancia de extensión que ya esté aprovisionada en pilas.

CloudFormation considera que las actualizaciones de versiones principales de las extensiones pueden contener cambios importantes y, por lo tanto, requiere que actualice manualmente a una nueva versión principal de una extensión.

Las extensiones publicadas por AWS están activadas de forma predeterminada para todas las cuentas y regiones en las que están disponibles y siempre utilizan la última versión disponible en cada Región de AWS.

Important

Dado que usted controla si las extensiones se actualizan a la última versión en su cuenta y cuándo, puede acabar con diferentes versiones de la misma extensión implementadas en diferentes cuentas y regiones.

Esto podría dar lugar a resultados inesperados al usar la misma plantilla, que contiene esa extensión, en esas cuentas y regiones.

Utilización de alias para hacer referencia a extensiones

No puede activar más de una extensión con un nombre de pila en una Cuenta de AWS y región determinadas. Dado que diferentes editores pueden ofrecer extensiones públicas con el mismo nombre de extensión, CloudFormation le permite especificar un alias para cualquier extensión pública de terceros que active.

Si especifica un alias para la extensión, CloudFormation trata el alias como el nombre del tipo de extensión dentro de la cuenta y la región. Debe usar el alias para hacer referencia a la extensión en las plantillas, las llamadas a la API y la consola de CloudFormation.

Los alias de extensión tienen que ser únicos en una cuenta y región determinadas. Puede activar el mismo recurso público varias veces en la misma cuenta y región, mediante alias de nombres de tipos diferentes.

Important

Si bien los alias de extensión solo deben ser únicos en una cuenta y región determinadas, recomendamos encarecidamente a los usuarios no asignar el mismo alias a diferentes

extensiones públicas de terceros en todas las cuentas y regiones. Esto podría generar resultados inesperados al usar una plantilla que contenga el alias de extensión en varias cuentas o regiones.

Comandos de la AWS CLI de uso común para trabajar con extensiones públicas

Los comandos comúnmente utilizados para trabajar con extensiones públicas incluyen:

- [activate-type](#) activar un módulo o un tipo de recurso público de terceros en su cuenta.
- [set-type-configuration](#) para especificar los datos de configuración de la extensión en su cuenta y para deshabilitar y habilitar enlaces.
- [list-types](#) para enumerar las extensiones en su cuenta.
- [describe-type](#) para devolver información detallada sobre una extensión específica o una versión de extensión específica, incluidos los datos de configuración actuales.
- [set-type-default-version](#) para especificar qué versión de una extensión es la versión predeterminada.
- [deactivate-type](#) desactivar un módulo o tipo de recurso público de terceros que se activó anteriormente en su cuenta.

Activación de una extensión pública de terceros en su cuenta

En el siguiente tema, se muestra cómo activar una extensión pública de terceros en su cuenta, de modo que se pueda utilizar en la cuenta y la región en la que se activó.

Note

Antes de continuar, confirme que ha creado la [rol de IAM](#) que usará con esta extensión.

Temas

- [Activación de una extensión pública \(consola\)](#)
- [Activación de una extensión pública \(AWS CLI\)](#)

Activación de una extensión pública (consola)

Siga los pasos de esta sección para utilizar la consola a fin de hacer lo siguiente:

- Activación de una extensión pública de terceros
- Especifique datos de configuración de extensión adicionales para su cuenta

Activación de una extensión pública para usarla en su cuenta

1. Inicie sesión en la Consola de administración de AWS y abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation>.
2. En la barra de navegación de la parte superior de la pantalla, elija la Región de AWS.
3. En el panel de navegación, en Registro, seleccione Extensiones públicas.
4. Use Filtrar para seleccionar el tipo de extensión y seleccione Terceros. (Las extensiones publicadas por AWS están activadas de manera predeterminada).
5. Seleccione la extensión y, a continuación, seleccione Activar.

Si hay varias versiones de una extensión disponibles, puede utilizar el menú Versión para seleccionar la versión de la extensión que desea activar. La versión predeterminada es la versión más actualizada.

6. Para Nombre de extensión, puede mantener seleccionada la opción Usar predeterminado o elegir Anular predeterminado y, a continuación, introducir el alias del tipo de extensión que desea utilizar con esta extensión. El alias debe seguir el formato recomendado para el tipo de extensión. Para obtener más información, consulte [Utilización de alias para hacer referencia a extensiones](#).
7. Si la extensión que va a activar es un enlace o un tipo de recurso, en ARN del rol de ejecución, especifique el rol de IAM que CloudFormation debe asumir al invocar la extensión. Para obtener más información, consulte [Configure un rol de ejecución con permisos de IAM y una política de confianza para el acceso a las extensiones públicas](#).
8. Para Configuración de registro, especifique la información de configuración de registro para una extensión, si lo desea. Por ejemplo:

```
{
  "logRoleArn": "arn:aws:iam::account:role/rolename",
  "logGroupName": "log-group-name"
}
```

No es necesario registrar la información de configuración, pero se recomienda para fines de depuración. Para usar la configuración de registro con enlaces, agregue la misma política de confianza que el rol de ejecución especificado, de modo que el rol de registro pueda escribir registros en el grupo de registros.

Los nombres de clave `logRoleArn` y `logGroupName` distinguen entre mayúsculas y minúsculas.

9. Para Control de versiones, Actualizaciones automáticas, seleccione cómo quiere recibir las actualizaciones.
 - On (Activado): se actualiza automáticamente a la última versión secundaria. Las versiones principales se actualizan manualmente.
 - Off (Desactivado): nunca se actualiza automáticamente a la última versión secundaria. Todas las versiones se actualizan manualmente.

Para obtener más información, consulte [Utilice automáticamente nuevas versiones de extensiones](#).

Si la extensión requiere configuración adicional, tiene la opción de especificar los datos de configuración ahora o después de que se haya activado la extensión.

Important

Si la extensión que está activando es un enlace, este paso es obligatorio. Debe especificar `ENABLED` para la propiedad `HookInvocationStatus`. Esta operación habilita las propiedades del enlace que se definen en la sección `properties` del esquema del enlace. Para obtener más información, consulte [Referencia a la sintaxis del esquema de la configuración del enlace](#) en la Guía del usuario de enlaces de CloudFormation.

Especificación de los datos de configuración

1. Para la Configuración, elija Configurar ahora y, a continuación, seleccione Activar la extensión.

CloudFormation mostrará la página `Configure extension` (Configurar extensión). Para ver el esquema de configuración actual de la extensión, asegúrese de que la opción `View configuration schema` (Ver esquema de configuración) esté activada.

2. En el cuadro de texto JSON de configuración, ingrese una cadena JSON que represente los datos de configuración que desea especificar para esta extensión. El JSON que especifique debe validarse con el esquema de configuración de la extensión.
3. Seleccione Configurar extensión.

Si prefiere configurar la extensión después de la activación, puede omitir este paso y proporcionar los datos de configuración más adelante.

1. Para la Configuración, elija Configurar más tarde y, a continuación, elija Activar la extensión.
2. Una vez activada la extensión, puede configurarla accediendo a la extensión desde la página de extensiones activadas y proporcionando los datos de configuración.

Activación de una extensión pública (AWS CLI)

Siga los pasos de esta sección para utilizar la AWS CLI a fin de hacer lo siguiente:

- Activación de una extensión pública de terceros
- Especifique datos de configuración de extensión adicionales para su cuenta

Activación de enlaces públicos

Al activar los enlaces en su cuenta, autoriza a un enlace a utilizar permisos definidos de su Cuenta de AWS. CloudFormation elimina los permisos no requeridos antes de pasar sus permisos al enlace. CloudFormation recomienda a los clientes o usuarios del enlace que revisen los permisos del enlace y sepan a qué permisos están permitidos los enlaces antes de habilitarlos en la cuenta.

Activación de un enlace público para usarlo en su cuenta (AWS CLI)

1. Obtenga el ARN para su enlace y guárdelo. Puede obtener el ARN de un enlace mediante Consola de administración de AWS o AWS CLI. Para obtener más información consulte () [Visualización de las extensiones disponibles y activadas en el registro de CloudFormation](#).

```
export HOOK_TYPE_ARN="arn:aws:cloudformation:us-west-2:123456789012:type/hook/  
Organization-Service-Hook/"
```

2. Utilice el comando [set-type-configuration](#) para especificar los datos de configuración. El JSON que pase para `--configuration` debe validarse con el esquema de configuración del

enlace. Para activar el enlace en todas las operaciones de pila, debe establecer la propiedad `HookInvocationStatus` en `ENABLED` en la sección `HookConfiguration`.

```
aws cloudformation set-type-configuration \
  --configuration '{"CloudFormationConfiguration":{"HookConfiguration":
{"HookInvocationStatus": "ENABLED", "FailureMode": "FAIL", "Properties":{}}}}' \
  --type-arn $HOOK_TYPE_ARN --region us-west-2
```

Para obtener más información acerca de las opciones de configuración de `HookConfiguration`, consulte [Referencia a la sintaxis del esquema de la configuración del enlace](#) en la Guía del usuario de enlaces de CloudFormation.

Activación de los módulos públicos y los tipos de recursos

Activación de una extensión pública para usarla en su cuenta (AWS CLI)

- Utilice el comando [activate-type](#) para activar la extensión y especifique si se actualizará de forma automática cada vez que se publique una nueva versión secundaria de la extensión.

En el ejemplo siguiente se especifica el nombre de recurso de Amazon (ARN) público de una extensión pública para activar para esta cuenta. Además, se especifica que CloudFormation actualiza la extensión cada vez que se publica una nueva versión secundaria.

```
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
```

Este comando devuelve un ARN de la extensión activada.

```
{
  "Arn": "arn:aws:cloudformation:us-west-2:123456789012:type/resource/My-
Resource-Example"
}
```

Actualización de la versión de una extensión pública (AWS CLI)

Utilice [activate-type](#) para volver a activar la extensión.

Utilice la opción `--version-bump` para especificar si se debe actualizar la extensión a la versión más reciente de MAJOR, la versión más reciente de MINOR.

```
aws cloudformation activate-type --type RESOURCE \  
--type-name Example::Test::1234567890abcdef0 \  
--type-name-alias Example::Test::Alias \  
--version-bump MAJOR --region us-west-2
```

Actualización de una extensión pública de terceros en su cuenta

Cuando activa una extensión pública de terceros, puede actualizar la mayoría de los detalles de la extensión desde su cuenta.

Actualización de una extensión pública en su cuenta (consola)

1. Inicie sesión en la Consola de administración de AWS y abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation>.
2. En la barra de navegación de la parte superior de la pantalla, elija la Región de AWS.
3. En el panel de navegación, en Registro de CloudFormation, seleccione Extensiones activadas.
4. Encuentre la extensión que desea actualizar y selecciónela. Para obtener más información, consulte [Visualización de las extensiones disponibles y activadas en el registro de CloudFormation](#).
5. En el menú Acciones, seleccione Editar y, a continuación, la opción de edición adecuada:
 - Para actualizar el esquema de configuración, consulte [Edición de los datos de configuración de las extensiones en su cuenta](#).
 - Para activar o desactivar las actualizaciones automáticas:
 1. Seleccione Editar actualizaciones automáticas.
 2. Elija Encendido o Apagado y, a continuación, elija Guardar. Para obtener más información, consulte [Utilice automáticamente nuevas versiones de extensiones](#).
 - Para actualizar el rol de ejecución:
 1. Seleccione Editar rol de ejecución.
 2. Especifique el ARN del rol de IAM que desea que CloudFormation utilice al invocar esta extensión y, a continuación, seleccione Guardar. Para obtener más información, consulte [Configure un rol de ejecución con permisos de IAM y una política de confianza para el acceso a las extensiones públicas](#).

- Para actualizar la configuración de registro:
 1. Seleccione Editar configuración de registro.
 2. Edite el JSON de configuración de registro y, a continuación, seleccione Guardar.

Desactivación de extensiones públicas de terceros en su cuenta

Cuando ya no necesite activar una extensión pública de terceros, siga los siguientes procedimientos para desactivarla en su cuenta.

Temas

- [Desactivación de una extensión pública en su cuenta \(consola\)](#)
- [Desactivación de una extensión pública en su cuenta \(AWS CLI\)](#)
- [Deshabilitación de un enlace en su cuenta \(AWS CLI\)](#)

Desactivación de una extensión pública en su cuenta (consola)

Desactivación de una extensión pública en su cuenta

1. Inicie sesión en la Consola de administración de AWS y abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation>.
2. En la barra de navegación de la parte superior de la pantalla, elija la Región de AWS.
3. En el panel de navegación, en Registro de CloudFormation, seleccione Extensiones activadas.
4. Encuentre la extensión que quiere desactivar y selecciónela. Para obtener más información, consulte [Visualización de las extensiones disponibles y activadas en el registro de CloudFormation](#).
5. En el menú Acciones, seleccione Desactivar.
6. Elija Desactivar.

Desactivación de una extensión pública en su cuenta (AWS CLI)

Use el siguiente comando [deactivate-type](#).

```
aws cloudformation deactivate-type --type MODULE \  
--type-name Example::Test::Type::MODULE \
```

```
--region us-west-2
```

Deshabilitación de un enlace en su cuenta (AWS CLI)

La deshabilitación de un enlace evita que este se ejecute en su Cuenta de AWS sin tener que eliminarlo.

Utilice el comando [set-type-configuration](#) y especifique HookInvocationStatus como DISABLED para deshabilitar un enlace.

En el siguiente ejemplo, se especifica la Región de AWS y el nombre de recurso de Amazon (ARN) del enlace que se deshabilitará.

```
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
```

Para obtener más información, consulte [Habilitación y deshabilitación de enlaces de CloudFormation](#) en la Guía del usuario de enlaces de CloudFormation.

Uso de extensiones privadas de terceros que se han compartido con usted

Para utilizar extensiones privadas de terceros que se han compartido con usted, primero debe registrarlas en CloudFormation, en las cuentas y regiones donde desea utilizarlas. Al registrar la extensión, se carga una copia de la extensión en el registro de CloudFormation de su cuenta y se activa. Una vez que haya registrado una extensión privada, aparecerá en el registro de CloudFormation para esa Cuenta de AWS y región, y podrá usarla en las plantillas de pila.

Temas

- [Permisos de IAM para registrar una extensión privada de terceros](#)
- [Comandos de la AWS CLI de uso común para trabajar con extensiones privadas](#)
- [Registro de una extensión privada de terceros en su cuenta](#)
- [Eliminación de extensiones privadas de terceros de la cuenta](#)

Permisos de IAM para registrar una extensión privada de terceros

Como parte del registro de una extensión privada, puede especificar un bucket de Amazon S3 que contiene el paquete de proyecto de la extensión. Este paquete contiene todos los archivos fuente necesarios para la extensión que desea registrar. El usuario que registra la extensión debe poder acceder al paquete del proyecto en el bucket de Amazon S3. Para hacerlo, el usuario debe tener permisos [GetObject](#) para el paquete de extensión.

Esto se cumple si utiliza el comando [register-type](#) de la AWS CLI o el comando [submit](#) de la CLI de CloudFormation.

Para obtener más información, consulte [Acciones, recursos y claves de condición de Amazon S3](#) en la Referencia de autorización de servicios.

Comandos de la AWS CLI de uso común para trabajar con extensiones privadas

Los comandos comúnmente utilizados para trabajar con extensiones privadas incluyen:

- [register-type](#) para registrar una extensión privada en su cuenta.
- [describe-type-registration](#) para devolver el estado actual de una solicitud de registro.
- [list-types](#) para enumerar las extensiones en su cuenta.
- [describe-type](#) para devolver información detallada sobre una extensión específica o una versión de extensión específica, incluidos los datos de configuración actuales.
- [set-type-configuration](#) para especificar los datos de configuración de la extensión en su cuenta y para deshabilitar y habilitar enlaces.
- [set-type-default-version](#) para especificar qué versión de una extensión es la versión predeterminada.
- [deregister-type](#) para eliminar una extensión privada o una versión de extensión de su cuenta.

Registro de una extensión privada de terceros en su cuenta

En este tema se explican los pasos para registrar una extensión privada de terceros que se haya compartido con usted a fin de que esté disponible para usarla en su cuenta.

 Note

Antes de continuar, confirme que tiene los requisitos [Permisos de IAM](#) para registrar una extensión privada.

Registro de una extensión privada que se comparte con usted (AWS CLI)

1. Busque el bucket de Amazon S3 que contiene el paquete de proyecto para la extensión privada que desea registrar en la cuenta.
2. Use el comando [register-type](#) para registrar la extensión privada en su cuenta.

Por ejemplo, el siguiente comando registra el tipo de recurso `My::Resource::Example` en la Cuenta de AWS especificada.

```
aws cloudformation register-type --type RESOURCE \  
  --type-name My::Resource::Example \  
  --schema-handler-package [s3 object path] --region us-west-2
```

`RegisterType` es una operación asíncrona y devuelve un token de registro que puede utilizar para hacer un seguimiento del progreso de la solicitud de registro.

```
{  
  "RegistrationToken": "f5525280-104e-4d35-bef5-8f1fexample"  
}
```

Si su extensión llama a las API de AWS como parte de su funcionalidad, debe crear un rol de ejecución de IAM que incluya los permisos necesarios para llamar a esas API de AWS y aprovisionar ese rol de ejecución en la cuenta. A continuación, puede especificar este rol de ejecución mediante la opción `--execution-role-arn`. CloudFormation asume entonces ese rol de ejecución para proporcionar las credenciales adecuadas a su tipo de recurso.

```
--execution-role-arn arn:aws:iam::123456789012:role/MyIAMRole
```

3. (Opcional) Utilice el token de registro con el comando [describe-type-registration](#) para realizar un seguimiento del progreso de su solicitud de registro.

Cuando CloudFormation completa la solicitud de registro, establece el estado de progreso de la solicitud en `COMPLETE`.

En el siguiente ejemplo, se utiliza el token de registro devuelto por el comando `describe-type-registration` anterior para devolver información de estado de registro.

```
aws cloudformation describe-type-registration \
  --registration-token f5525280-104e-4d35-bef5-8f1fexample \
  --region us-west-2
```

El comando devuelve el resultado siguiente.

```
{
  "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

Si la extensión que está registrando es un enlace, el paso siguiente es obligatorio. Debe especificar `ENABLED` para la propiedad `HookInvocationStatus`. Esta operación habilita las propiedades del enlace que se definen en la sección `properties` del esquema del enlace. Para obtener más información, consulte [Referencia a la sintaxis del esquema de la configuración del enlace](#) en la Guía del usuario de enlaces de CloudFormation.

Especificar los datos de configuración para un enlace (AWS CLI)

1. Obtenga el ARN para su enlace y guárdelo. Puede obtener el ARN de un enlace mediante Consola de administración de AWS o AWS CLI. Para obtener más información consulte ([Visualización de las extensiones disponibles y activadas en el registro de CloudFormation](#)).

```
export HOOK_TYPE_ARN="arn:aws:cloudformation:us-west-2:123456789012:type/hook/Organization-Service-Hook"
```

2. Utilice el comando [set-type-configuration](#) para especificar los datos de configuración. El JSON que pase para `--configuration` debe validarse con el esquema de configuración del enlace.

Para activar el enlace, debe establecer la propiedad `HookInvocationStatus` en `ENABLED` en la sección `HookConfiguration`.

```
aws cloudformation set-type-configuration \
  --configuration '{"CloudFormationConfiguration":{"HookConfiguration":
{"HookInvocationStatus": "ENABLED", "FailureMode": "FAIL", "Properties":{}}}}' \
  --type-arn $HOOK_TYPE_ARN --region us-west-2
```

Para obtener más información, consulte [Referencia a la sintaxis del esquema de la configuración del enlace](#) en la Guía del usuario de enlaces de CloudFormation.

Eliminación de extensiones privadas de terceros de la cuenta

Para eliminar una extensión privada de terceros o una versión de extensión, use el comando [deregister-type](#).

Puede anular el registro de una versión de extensión específica o de la extensión en su totalidad. Para anular el registro de una extensión, debe anular el registro de forma individual todas las versiones registradas de esa extensión. Si una extensión solo tiene una versión registrada, la anulación del registro de esa versión resultará en la anulación del registro de la propia extensión. No puede anular el registro de la versión predeterminada de una extensión, a menos que sea la única versión registrada de esa extensión, en cuyo caso también se anulará el registro de la propia extensión.

Warning

No se puede deshacer la anulación del registro de una extensión privada. Esta acción permitirá:

- Hacer que la extensión no se pueda utilizar en todas las operaciones de CloudFormation.
- Provocar errores en las futuras actualizaciones de la pila que utilicen esta extensión (para tipos de recursos y módulos). Si bien puede volver a registrar la extensión de forma privada más adelante, esto podría provocar errores si CloudFormation se basa en una versión anterior.

Antes de continuar, utilice los comandos [list-stacks](#) y [get-template](#) para comprobar que ninguna pila activa utilice esta extensión.

Ejemplos de comandos para anular el registro de extensiones

En esta sección se proporcionan ejemplos que muestran las diferentes formas de anular el registro de extensiones privadas.

Cómo anular el registro por nombre de tipo

Use el comando [deregister-type](#) con las opciones `--type` y `--type-name` para anular el registro de la extensión.

```
aws cloudformation deregister-type \  
  --type MODULE \  
  --type-name My::S3::SampleBucket::MODULE
```

Cómo anular el registro por versión y nombre de tipo

Para anular el registro de una versión específica de la extensión, especifique la opción `--version-id` en el comando.

```
aws cloudformation deregister-type \  
  --type MODULE \  
  --type-name My::S3::SampleBucket::MODULE \  
  --version-id 00000001
```

Tip

Para establecer primero una versión diferente de la extensión como predeterminada, utilice el comando [set-type-default-version](#).

Cómo anular el registro por ARN

Use la opción `--arn` y especifique el ARN de su extensión para anular el registro.

```
aws cloudformation deregister-type \  
  --arn arn:aws:cloudformation:us-west-2:123456789012:type/resource/Organization-Service-Resource
```

Edición de los datos de configuración de las extensiones en su cuenta

En este tema se proporcionan instrucciones sobre cómo editar los datos de configuración de las extensiones de su cuenta dentro de una región específica. Las extensiones pueden incluir propiedades de configuración que aplican a todas las instancias de la extensión para una cuenta y región determinadas. El autor de la extensión las define en la configuración de definición de la extensión. Si hay propiedades obligatorias en la definición de configuración de la extensión, debe especificarlas antes de poder usar la extensión en su cuenta y región.

Para obtener más información sobre cómo se definen las opciones de definición al desarrollar una extensión, consulte la siguiente documentación.

- [Referencia a la sintaxis del esquema de la configuración del enlace](#)
- [Defining the account-level configuration of an extension \(Definición de la configuración de cuenta de una extensión\)](#)

Temas

- [Permisos necesarios para usar referencias dinámicas](#)
- [Edición de los datos de configuración de una extensión \(consola\)](#)
- [Edición de los datos de configuración de una extensión \(AWS CLI\)](#)

Permisos necesarios para usar referencias dinámicas

Si la configuración de datos incluye referencias dinámicas a valores almacenados en AWS Systems Manager o AWS Secrets Manager, cualquier rol que se use para aprovisionar el tipo (por ejemplo, al crear o actualizar una pila) debe tener los permisos adecuados para recuperar ese valor. En concreto:

- Si la configuración de datos contiene un parámetro almacenado en el almacén de parámetros de AWS Systems Manager, el usuario o rol que se use para aprovisionar el tipo debe tener permisos para llamar a [GetParameter](#).
- Si la configuración de datos contiene un secreto almacenado en AWS Secrets Manager, el usuario o rol que se use para aprovisionar el tipo debe tener permisos para llamar a [GetSecretValue](#).

Para obtener más información, consulte [Obtención de valores almacenados en otros servicios con referencias dinámicas](#).

Edición de los datos de configuración de una extensión (consola)

Siga los pasos de esta sección para utilizar la consola a fin de hacer lo siguiente:

- Visualización de los datos de configuración actuales de una extensión
- Actualización de los datos de configuración de la extensión de su cuenta

Visualización de los datos de configuración actuales de una extensión

1. Inicie sesión en la Consola de administración de AWS y abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation>.
2. En la barra de navegación de la parte superior de la pantalla, elija la Región de AWS.
3. En el panel de navegación, en Registro de CloudFormation, seleccione Extensiones activadas.
4. Encuentre la extensión que desea ver. Para obtener más información, consulte [Visualización de las extensiones disponibles y activadas en el registro de CloudFormation](#).
5. Elija la extensión para ver sus detalles.
6. En la página de detalles de la extensión, elija la pestaña Configuration (Configuración).
7. Expanda la pestaña Configuration schema (Esquema de configuración) para ver el esquema de configuración definido para la extensión.
8. Expanda la pestaña Configuration (Configuración) para ver la configuración actual que ha establecido para esta extensión.

Actualización de los datos de configuración de una extensión

1. En la página de detalles de la extensión, en la pestaña Configuración, seleccione Editar configuración.

Como alternativa, desde Acciones, seleccione Editar y, a continuación, seleccione Editar configuración.

CloudFormation mostrará la página Configure extension (Configurar extensión). Asegúrese de que View configuration schema (Ver esquema de configuración) esté activado para ver el esquema de definición de configuración actual de la extensión.

2. En el cuadro de texto Configuration JSON (JSON de configuración), ingrese una cadena JSON que represente el esquema de configuración que desea especificar para esta extensión. Debe validarse con respecto al esquema definido en Configuration schema (Esquema de configuración).
3. Seleccione Configurar extensión.

Edición de los datos de configuración de una extensión (AWS CLI)

Siga los pasos de esta sección para utilizar la AWS CLI a fin de hacer lo siguiente:

- Visualización de los datos de configuración actuales de una extensión
- Actualización de los datos de configuración de la extensión de su cuenta

Visualización de los datos de configuración actuales de una extensión

- Use el comando [describe-type](#) para devolver información detallada acerca de la extensión. El elemento ConfigurationSchema de la salida contiene la definición de configuración actual de la extensión en una región determinada.

Como alternativa, consulte el comando [batch-describe-type-configurations](#) para devolver datos de configuración sobre varias extensiones.

Actualización de los datos de configuración de una extensión

- Utilice el comando [set-type-configuration](#) para especificar los datos de configuración. El JSON que pase para --configuration debe validarse con el esquema de configuración de la extensión.

En el siguiente ejemplo, el comando set-type-configuration especifica los datos de configuración `{"CredentialKey": "testUserCredential"}` de la opción --configuration.

```
aws cloudformation set-type-configuration --type RESOURCE \  
  --type-name My::Resource::Example \  
  --configuration-alias default \  
  --configuration {"CredentialKey": "testUserCredential"} \  
  --region us-west-2
```

Registro de tipos de recursos en AWS Config

Puede especificar que AWS Config realice un seguimiento automático de los tipos de recursos privados y registre los cambios en dichos recursos como elementos de configuración. Esto le permite ver el historial de configuración de estos tipos de recursos privados, así como escribir reglas de Reglas de AWS Config para verificar las prácticas recomendadas de configuración. Se requiere AWS Config para la extensión de enlace.

Para que AWS Config realice un seguimiento automático de los tipos de recursos privados:

- Administre los recursos a través de CloudFormation. Esto incluye la realización de todas las operaciones de creación, actualización y eliminación de recursos a través de CloudFormation.

Note

Si utiliza un rol de IAM para realizar las operaciones de pila, ese rol de IAM debe tener permiso para llamar a las siguientes acciones de AWS Config:

- [PutResourceConfig](#)
- [DeleteResourceConfig](#)

- Configure AWS Config para registrar todos los tipos de recursos. Para obtener más información, consulte el [Registro de configuraciones para recursos de terceros mediante la AWS CLI](#) en la Guía para desarrolladores de AWS Config.

Note

AWS Config no admite la grabación de recursos privados que contienen propiedades definidas como obligatorias y de solo escritura.

Por diseño, las propiedades de recurso definidas como solo escritura no se devuelven en el esquema utilizado para crear el elemento de configuración AWS Config. Debido a esto, si se incluye una propiedad definida como solo escritura y obligatoria, se producirá un error en la creación del elemento de configuración, ya que no se presentará una propiedad obligatoria. Para ver el esquema que se utilizará para crear el elemento de configuración, puede revisar la propiedad `schema` de la acción [DescribeType](#).

Para obtener más información sobre los elementos de configuración, consulte [Configuration items](#) (Elementos de configuración) en la Guía para desarrolladores de AWS Config.

Impedir que se registren propiedades confidenciales en un elemento de configuración

El tipo de recurso puede contener propiedades que considere información confidencial, como contraseñas, secretos u otros datos confidenciales, que no desea que se registren como parte del elemento de configuración. Para evitar que se registre una propiedad en el elemento de configuración, puede incluir esa propiedad en la lista `writeOnlyProperties` del esquema de tipo de recurso. El usuario puede especificar las propiedades de recurso mostradas como `writeOnlyProperties`, pero no se devolverán por una solicitud `read` ni `list`.

Para obtener más información, consulte [writeOnlyProperties](#) en la Guía del usuario de la CLI de CloudFormation.

Entrega continua con CodePipeline

La entrega continua es una práctica de desarrollo de software en la que los cambios del código se crean, comprueban y preparan automáticamente para lanzarlos en el entorno de producción. Con CloudFormation y CodePipeline, puede implementar entrega continua para construir y probar de forma automática cambios en las plantillas de CloudFormation antes de promoverlas a pilas de producción. Este proceso de lanzamiento permite realizar cambios de manera rápida y fiable a la infraestructura de AWS.

Por ejemplo, puede crear un flujo de trabajo que crea una pila de prueba automáticamente cuando envía una plantilla actualizada a un repositorio del código. Después de que CloudFormation construye la pila de prueba, puede probarla y luego decidir si desea aplicar los cambios a una pila de producción. Para obtener más información sobre los beneficios de la entrega continua, consulte [¿Qué es la entrega continua?](#)

Utilice CodePipeline para establecer un flujo de trabajo de entrega continua mediante la creación de una canalización para las pilas de CloudFormation. CodePipeline cuenta con integración incorporada con CloudFormation, lo que le permite especificar acciones específicas de CloudFormation, como crear, actualizar o eliminar una pila, dentro de una canalización. Para obtener más información sobre CodePipeline, consulte la [Guía del usuario de AWS CodePipeline](#).

Temas

- [Tutorial: Creación de una canalización para pilas de prueba y pilas de producción](#)
- [CloudFormationReferencia de propiedades de configuración de](#)
- [CloudFormationArtefactos de](#)
- [Uso de funciones de anulación de parámetros con canalizaciones de CodePipeline](#)

Tutorial: Creación de una canalización para pilas de prueba y pilas de producción

Imagine un proceso de publicación en el que envía una plantilla de CloudFormation que CloudFormation utiliza a continuación para crear una pila de prueba de forma automática. Después de revisar la pila de prueba, puede obtener una vista previa de cómo modificarán sus cambios la pila de producción y, a continuación, decidir si desea implementarlos. Para aplicar este flujo de trabajo, podría usar CloudFormation para crear la pila de prueba, eliminar la pila de prueba, crear un conjunto

de cambios y, a continuación, ejecutar el conjunto de cambios. No obstante, con cada acción deberá interactuar manualmente con CloudFormation. En este tutorial, crearemos una canalización de CodePipeline que automatiza muchas de estas acciones, lo que le ayudará a conseguir un flujo de trabajo de entrega continua con sus pilas de CloudFormation.

Requisitos previos

En este tutorial, se presupone que ha utilizado CodePipeline y CloudFormation, y sabe cómo funcionan las canalizaciones, las pilas y las plantillas de AWS CloudFormation. Para obtener más información sobre CodePipeline, consulte la [Guía del usuario de AWS CodePipeline](#). También debe contar con un bucket de Amazon S3 en la misma Región de AWS en la que creará la canalización.

Important

La plantilla de ejemplo de WordPress crea una instancia de EC2 que requiere una conexión a Internet. Compruebe que dispone de una VPC y subred predeterminadas que permitan el tráfico a Internet.

Información general del tutorial

En este tutorial se crea una canalización para un sitio de WordPress de ejemplo en una pila. La canalización se divide en tres fases. Cada fase debe contener al menos una acción, que es una tarea que la canalización realiza sobre los artefactos (la entrada que usted especifica). Una fase organiza las acciones en una canalización. CodePipeline debe completar todas las acciones de una fase antes de que esta procese nuevos artefactos; por ejemplo, cuando haya enviado nuevas entradas para volver a ejecutar la canalización.

Al final de este tutorial tendrá una canalización que aplicará el siguiente flujo de trabajo:

1. La primera fase de la canalización recupera un artefacto de origen (una plantilla de CloudFormation y sus archivos de configuración) de un repositorio.

Preparará un artefacto que incluirá una plantilla de WordPress de ejemplo y lo cargará en un bucket de S3.

2. En la segunda fase, la canalización crea una pila de prueba y, a continuación, espera su aprobación.

Después de revisar la pila de prueba, puede elegir entre continuar con la canalización original o crear y enviar otro artefacto para realizar cambios. Si da su aprobación, esta fase elimina la pila de prueba y, a continuación, la canalización continúa con la siguiente fase.

3. En la tercera fase, la canalización crea un conjunto de cambios para una pila de producción y, a continuación, espera su aprobación.

En la ejecución inicial, no habrá una pila de producción. El conjunto de cambios muestra todos los recursos que CloudFormation creará. Si da su aprobación, esta fase ejecuta el conjunto de cambios y crea la pila de producción.

Note

CloudFormation es un servicio gratuito. Sin embargo, se le cobrará por los recursos de AWS que incluya en la pila, como la instancia de EC2, aplicando la tarifa actual para cada una de ellos. Para obtener más información sobre los precios de AWS, consulte la página de detalles correspondiente a cada producto en <http://aws.amazon.com>.

Paso 1: Editar el artefacto y cargarlo en un bucket de S3

Antes de crear la canalización, debe configurar el repositorio de origen y los archivos. CodePipeline copia estos archivos de origen en el [almacén de artefactos](#) de su canalización y, a continuación, los utiliza para llevar a cabo las acciones de la canalización, como, por ejemplo, crear una pila de CloudFormation.

Cuando se usa Amazon Simple Storage Service (Amazon S3) como repositorio de origen, CodePipeline requiere que comprima los archivos de origen antes de cargarlos a un bucket de S3. El archivo comprimido es un artefacto de CodePipeline que puede contener una plantilla de CloudFormation, un archivo de configuración de plantilla o ambos. En la guía proporcionamos un artefacto que contiene como ejemplo una plantilla de WordPress y dos archivos de configuración de plantilla. Los dos archivos de configuración especifican los valores de los parámetros de la plantilla de WordPress. CodePipeline los utiliza cuando crea las pilas de WordPress. Un archivo contiene los valores de los parámetros para una pila de prueba y el otro para una pila de producción. Tendrá que editar los archivos de configuración, por ejemplo, para especificar el nombre de un par de claves de EC2 de su propiedad. Para obtener más información acerca de los artefactos, consulte [CloudFormationArtefactos de](#).

Después de crear el artefacto, deberá cargarlo en un bucket de S3.

Para editar y cargar el artefacto

1. Descargue y abra el artefacto de ejemplo: <https://s3.amazonaws.com/cloudformation-examples/user-guide/continuous-deployment/wordpress-single-instance.zip>.

El artefacto contiene tres archivos:

- La plantilla de WordPress de ejemplo: `wordpress-single-instance.yaml`
 - El archivo de configuración de plantilla para la pila de prueba: `test-stack-configuration.json`
 - El archivo de configuración de plantilla para la pila de producción: `prod-stack-configuration.json`
2. Extraiga todos los archivos y, a continuación, utilice cualquier editor de texto para modificar los archivos de configuración de la plantilla.

Abra los archivos de configuración para comprobar que contienen los pares clave-valor que se asignarán a los parámetros de la plantilla de WordPress. Los archivos de configuración especifican los valores de los parámetros que la canalización utiliza al crear las pilas de prueba y de producción.

Edite el archivo `test-stack-configuration.json` para especificar los valores de los parámetros para la pila de prueba y el archivo `prod-stack-configuration.json` para la pila de producción.

- Cambie los valores de las claves `DBPassword` y `DBRootPassword` por contraseñas que sirvan para iniciar sesión en la base de datos de WordPress. Siguiendo la definición de la plantilla de WordPress, los valores de los parámetros solo pueden contener caracteres alfanuméricos.
 - Cambie el valor de la clave `KeyName` por un nombre de par de claves de EC2 existente en la región en la que creará la canalización.
3. Agregue los archivos de configuración modificados al archivo (`.zip`) del artefacto original sustituyendo los archivos duplicados.

Ahora tiene un artefacto personalizado que puede cargar en un bucket de S3.

4. [Cargue el artefacto en un bucket de S3 de su propiedad.](#)

Observe la ubicación del archivo. Tendrá que especificarla después al crear la canalización.

Notas sobre el artefacto y el bucket de S3:

- Utilice un bucket que se encuentre en la misma región de AWS en la que creará la canalización.
- CodePipeline requiere que el bucket tenga el [control de versiones habilitado](#).
- También puede utilizar como repositorio de origen servicios que no requieren comprimir los archivos antes de cargarlos, como GitHub o CodeCommit.
- Los artefactos pueden contener información confidencial, por ejemplo contraseñas. Limite el acceso para que solo los usuarios autorizados puedan ver el archivo. Al hacerlo, asegúrese de que CodePipeline siga teniendo acceso al archivo. Por ejemplo, si carga el artefacto en un bucket de S3, utilice [políticas de bucket de S3 o políticas de usuario](#) para restringir el acceso.

Ahora tiene un artefacto que CodePipeline puede incluir en la canalización. En el siguiente paso, especificará la ubicación del artefacto y creará la canalización de WordPress.

Paso 2: Crear la pila de la canalización

Para crear la canalización de WordPress, utilizará una plantilla de CloudFormation de ejemplo. Además de crear la canalización, la plantilla configura roles del servicio de AWS Identity and Access Management (IAM) para CodePipeline y CloudFormation, un bucket de S3 para el almacén de artefactos de CodePipeline y un tema de Amazon Simple Notification Service (Amazon SNS) al que la canalización envía notificaciones, como las relacionadas con comentarios. La plantilla de ejemplo facilita las tareas de aprovisionamiento y configuración de estos recursos en una única pila de CloudFormation.

Para obtener más detalles sobre la configuración de la canalización, consulte [Operaciones que realiza la canalización](#).

Important

La plantilla de ejemplo de WordPress crea una instancia de EC2 que requiere una conexión a Internet. Compruebe que su VPC y subred predeterminadas permiten dirigir tráfico a Internet.

Para crear la pila de la canalización

1. Descargue la plantilla de ejemplo en <https://s3.amazonaws.com/cloudformation-examples/user-guide/continuous-deployment/basic-pipeline.yml>. Guárdela en su equipo.
2. Abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation/>.
3. Elija una región de AWS compatible con CodePipeline y CloudFormation.

Para obtener más información, consulte [Puntos de conexión y cuotas de CodePipeline](#) y [Puntos de conexión y cuotas de CloudFormation](#) en Referencia general de AWS.

4. Seleccione Creación de pila.
5. En Specify template (Especificar plantilla), elija Upload a template file (Cargar un archivo de plantilla) y, a continuación, elija la plantilla que acaba de descargar, `basic-pipeline.yml`.
6. Elija Siguiente.
7. Para Stack name (Nombre de pila), escriba `sample-WordPress-pipeline`.
8. En la sección Parameters (Parámetros) especifique los valores siguientes para los parámetros y, a continuación, elija Next (Siguiente). Al configurar los parámetros de la pila, si mantuvo los mismos nombres para la plantilla de WordPress y sus archivos de configuración, puede utilizar los valores predeterminados. En caso contrario, especifique los nombres de archivo que utilizó.

PipelineName

Nombre de la canalización, por ejemplo `WordPress-test-pipeline`.

S3 Bucket

Nombre del bucket de S3 donde guardó el artefacto (archivo `.zip`).

SourceS3Key

El nombre de archivo de su artefacto. Si ha guardado el artefacto en una carpeta, inclúyala como parte del nombre de archivo, como `folder/subfolder/wordpress-single-instance.zip`.

Correo electrónico

Dirección de correo electrónico a la que CodePipeline envía la notificación de la canalización, por ejemplo, `myemail@example.com`.

9. Para este tutorial, no tiene que añadir etiquetas ni especificar una configuración avanzada, así que elija Next (Siguiente).

10. Asegúrese de que el nombre de la pila y la URL de la plantilla sean correctos y elija **Create stack** (Crear pila).
11. Para confirmar que es consciente de que CloudFormation podría crear recursos de IAM, seleccione la casilla de verificación.

Podría llevarle varios minutos a AWS CloudFormation crear la pila. Para monitorizar el progreso, vea los eventos de la pila. Para obtener más información, consulte [Monitorizar el progreso de la pila](#).

Una vez creada la pila, CodePipeline inicia la nueva canalización. Para ver el estado, consulte la [consola de CodePipeline](#). En la lista de canalizaciones, elija WordPress-test-pipeline.

Operaciones que realiza la canalización

En esta sección se explican las tres fases de la canalización a través de fragmentos de código de la plantilla de la canalización de WordPress de ejemplo.

Fase 1: Origen

La primera fase de la canalización es una fase de origen en la que usted especifica la ubicación de su código fuente. Cada vez que envíe una revisión a esta ubicación, CodePipeline volverá a ejecutar la canalización.

El código fuente se encuentra en un bucket de S3 y se identifica mediante su nombre de archivo. Especifiqué estos valores para los parámetros de entrada al crear la pila de la canalización. Para permitir usar el artefacto de origen en las fases posteriores, el fragmento especifica la propiedad `OutputArtifacts`, con el nombre `TemplateSource`. Para utilizar este artefacto en fases posteriores, especifique `TemplateSource` como artefacto de entrada.

```
- Name: S3Source
  Actions:
    - Name: TemplateSource
      ActionTypeId:
        Category: Source
        Owner: AWS
        Provider: S3
        Version: '1'
      Configuration:
        S3Bucket: !Ref 'S3Bucket'
        S3ObjectKey: !Ref 'SourceS3Key'
      OutputArtifacts:
```

```
- Name: TemplateSource
```

Fase 2: TestStage

En la fase TestStage, la canalización crea la pila de prueba, espera a su aprobación y, a continuación, elimina la pila de prueba.

Para la acción CreateStack, la canalización utiliza el archivo de configuración de prueba y la plantilla de WordPress para crear la pila de prueba. Ambos archivos se encuentran en el artefacto de entrada TemplateSource, que se trae de la fase de origen. El fragmento de código utiliza el modo de acción REPLACE_ON_FAILURE. Si se produce algún error durante la creación de la pila, la canalización la sustituye para que no sea necesario limpiar o resolver problemas de la pila antes de poder volver a ejecutar la canalización. Este modo de acción es útil para una iteración rápida en pilas de prueba. Para la propiedad RoleArn, el valor es un rol del servicio AWS CloudFormation declarado en otro lugar de la plantilla.

La acción ApproveTestStack detiene la canalización y envía una notificación a la dirección de correo electrónico que especificó al crear la pila de la canalización. Mientras la canalización está en pausa, puede comprobar la pila de prueba de WordPress y sus recursos. Uso de CodePipeline para [aprobar o rechazar](#) esta acción. La propiedad CustomData incluye una descripción de la acción que va a aprobar, que la canalización añade al correo electrónico de notificación.

Después de aprobar esta acción, CodePipeline pasa a la acción DeleteTestStack y elimina la pila de prueba de WordPress y sus recursos.

```
- 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'

```

Fase 3: ProdStage

La fase ProdStage de la canalización crea un conjunto de cambios en la pila de producción existente, espera a su aprobación y, a continuación, ejecuta el conjunto de cambios.

Un conjunto de cambios proporciona una vista previa de todas las modificaciones que AWS CloudFormation hará a su pila de producción antes de su aplicación. En la primera ejecución de la canalización, no habrá una pila de producción en ejecución. El conjunto de cambios muestra las acciones que AWS CloudFormation realizó al crear la pila de prueba. Para crear el conjunto de cambios, la acción CreateChangeSet utiliza el ejemplo de plantilla de WordPress y la configuración de plantilla para producción del artefacto de entrada TemplateSource.

De forma similar a la fase anterior, la acción ApproveChangeSet detiene la canalización y envía una notificación por correo electrónico. Mientras la canalización está en pausa, puede ver el conjunto de cambios para comprobar todas las modificaciones propuestas a la pila de producción de WordPress. Use CodePipeline para [aprobar o rechazar](#) esta acción y así continuar o detener la canalización, respectivamente.

Después de aprobar esta acción, `ExecuteChangeSet` ejecuta el conjunto de cambios, de forma que AWS CloudFormation realice todas las acciones descritas en él. Para la ejecución inicial, AWS CloudFormation crea la pila de producción de WordPress. En las ejecuciones posteriores, AWS CloudFormation actualiza la pila.

```
- 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'
```

Paso 3: Visualizar la pila de WordPress

Cuando CodePipeline recorre la canalización, utiliza CloudFormation para crear pilas de prueba y de producción. Para ver el estado de estas pilas y su salida, utilice la consola de CloudFormation.

Para ver una pila

1. Abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation/>.
2. Dependiendo de si la canalización se encuentra en la fase de prueba o de producción, elija la pila Test-MyWordPressSite o la pila Prod-MyWordPressSite.
3. Para comprobar el estado de la pila, vea los [eventos](#) de la pila.

Si la pila se encuentra en estado de error, observe el motivo del estado para encontrar el error de la pila. Corrija el error y, a continuación, vuelva a ejecutar la canalización. Si la pila se encuentra en el estado CREATE_COMPLETE, observe sus salidas para determinar la URL del sitio de WordPress.

Ha utilizado CodePipeline con éxito para crear un flujo de trabajo de entrega continua para un sitio de WordPress de ejemplo. Si envía cambios al bucket de S3, CodePipeline detectará automáticamente la nueva versión y volverá a ejecutar la canalización. Este flujo de trabajo facilita el envío y la comprobación de los cambios antes de aplicarlos al sitio de producción.

Paso 4: limpie los recursos

Para asegurarse de que no se le cobra por servicios no deseados, elimine los recursos.

Important

Elimine las pilas de prueba y producción de WordPress antes de eliminar la pila de la canalización. La pila de la canalización contiene un rol de servicio necesario para eliminar las pilas de WordPress. Si ha eliminado primero la pila de la canalización, puede asociar el nombre de recurso de Amazon (ARN) de otro rol de servicio a las pilas de WordPress y, a continuación, eliminarlas.

Para eliminar objetos del almacén de artefactos

1. Abra la consola de Amazon S3 en <https://console.aws.amazon.com/s3>.
2. Elija el bucket de S3 que CodePipeline ha utilizado como almacén de artefactos de la canalización.

El nombre del bucket tiene este formato: *stackname-artifactstorebucket-id*. Si ha seguido este tutorial, el nombre del bucket puede ser similar al ejemplo siguiente: sample-WordPress-pipeline-artifactstorebucket-12345abcd12345.

3. Elimine todos los objetos del bucket de S3 del almacén de artefactos.

Al eliminar la pila de la canalización en el paso siguiente, el bucket debe estar vacío. De lo contrario, CloudFormation no podrá eliminar el bucket.

Para eliminar pilas

1. Desde la consola de CloudFormation, elija la pila que desea eliminar.

Si las pilas de WordPress creadas por la canalización siguen en ejecución, elíjalas en primer lugar. De forma predeterminada, los nombres de las pilas son Test-MyWordPressSite y Prod-MyWordPressSite.

Si ya ha eliminado las pilas de WordPress, elija la pila sample-WordPress-pipeline.

2. Elija Actions (Acciones) y, a continuación, elija Delete Stack (Eliminar pila).
3. En el mensaje de confirmación, elija Yes, Delete (Sí, eliminar).

CloudFormation elimina la pila y todos sus recursos, como la instancia EC2, el tema de notificación, el rol de servicio y la canalización.

Ahora que sabe cómo crear un flujo de trabajo de CloudFormation básico con CodePipeline, puede utilizar la plantilla y los artefactos de ejemplo como punto de partida para crear los suyos.

Véase también

Los recursos relacionados siguientes pueden serle de ayuda cuando trabaje con estos parámetros.

- Para obtener más información sobre los parámetros de acción de CloudFormation en CodePipeline, consulte la [referencia de configuración de acciones de CloudFormation](#) en la Guía del usuario de AWS CodePipeline.
- Para ver valores de plantilla de ejemplo según el proveedor de acción, como para el campo `Owner` o los campos `configuration`, consulte la [Action structure reference](#) (Referencia de estructura de acciones) en la Guía del usuario de AWS CodePipeline.
- Para descargar las plantillas de la pila de canalización de ejemplo en formato YAML o JSON, consulte el [Tutorial: Creación de una canalización con CloudFormation](#) en la Guía del usuario de AWS CodePipeline.

CloudFormationReferencia de propiedades de configuración de

Al crear una canalización de CodePipeline, agrega una acción `Deploy` a la canalización con CloudFormation como proveedor. A continuación, debe especificar qué acción de CloudFormation invoca la canalización y la configuración de la acción. Este tema describe las propiedades de configuración de CloudFormation. Para especificar propiedades, puede utilizar la consola de CodePipeline o crear un objeto JSON a fin de usarlo para las plantillas de la AWS CLI, la API de CodePipeline o CloudFormation.

Temas

- [Propiedades de configuración \(consola\)](#)
- [Propiedades de configuración \(objeto JSON\)](#)
- [Véase también](#)

Propiedades de configuración (consola)

La [consola](#) de CodePipeline muestra las propiedades de configuración e indica las propiedades necesarias en función del modo de acción que elija.

Note

Al crear una canalización, puede especificar solo los modos de acción `Create or update a stack` (Crear o actualizar una pila) o `Create or replace a change set` (Crear o reemplazar un conjunto de cambios). Además, las propiedades de la sección `Advanced` (Avanzadas) solo están disponibles al editar una canalización.

Action mode (Modo de acción)

La acción de CloudFormation que CodePipeline invoca al procesar la fase asociada. Elija uno de los siguientes modos de acción:

- **Create or replace a change set** (Crear o reemplazar un conjunto de cambios) crea el conjunto de cambios si no existe, basándose en el nombre de la pila y la plantilla que envíe. Si el conjunto de cambios ya existe, CloudFormation lo elimina y luego crea otro nuevo.
- **Create or update a stack** (Crear o reemplazar una pila) crea la pila si la pila especificada no existe. Si la pila existe, CloudFormation lo actualiza. Utilice esta acción para actualizar pilas existentes. CodePipeline no sustituirá la pila.
- **Delete a stack** (Eliminar una pila) elimina una pila. Si especifica una pila que no existe, la acción se realiza de forma satisfactoria sin eliminar una pila.
- **Execute a change set** (Ejecutar un conjunto de cambios) ejecuta un conjunto de cambios.
- **Replace a failed stack** (Reemplazar una pila con errores) crea la pila si la pila especificada no existe. Si la pila existe y está en un estado de error (notificado como `ROLLBACK_COMPLETE`, `ROLLBACK_FAILED`, `CREATE_FAILED`, `DELETE_FAILED` o `UPDATE_ROLLBACK_FAILED`), CloudFormation elimina la pila y luego crea otra nueva. Si la pila no está en un estado de error, CloudFormation lo actualiza. Utilice esta acción para sustituir pilas con error sin recuperarlos ni solucionar sus problemas. Por lo general, debería elegir este modo para las pruebas.

Nombre de pila

El nombre que se asocia a una pila existente o a una pila que desea crear. El nombre debe ser único en la región AWS en la que está creando la pila.

Note

El nombre de una pila solo puede contener caracteres alfanuméricos (con distinción de mayúsculas y minúsculas) y guiones. Debe comenzar por una letra y no puede tener más de 128 caracteres.

Cambiar nombre de conjunto

El nombre de un conjunto de cambios existentes o un nuevo conjunto de cambios que desee crear para la pila especificada.

Template

La ubicación de un archivo de plantilla de CloudFormation, que sigue el formato

ArtifactName::TemplateName.

Template configuration (Configuración de plantilla)

La ubicación de un archivo de configuración de plantilla, que sigue el formato

ArtifactName::TemplateConfigurationFileName. El archivo de configuración de la plantilla puede contener los valores de los parámetros de la plantilla, una política de pila y etiquetas. Si incluye información confidencial, como contraseñas, restrinja el acceso a este archivo. Para obtener más información, consulte [CloudFormationArtefactos de](#).

Capacidades

Respecto a las pilas que contengan determinados recursos, reconozca explícitamente que CloudFormation podría crear o actualizar dichos recursos. Por ejemplo, debe especificar `CAPABILITY_IAM` si la plantilla de su pila contiene recursos de AWS Identity and Access Management (IAM). Para obtener más información, consulte los parámetros [CreateStack](#) de solicitud de operación de API.

Si dispone de recursos de IAM en su plantilla de pila, debe especificar esta propiedad.

Puede especificar más de una capacidad.

Nombre del rol

El nombre del rol de servicio IAM que CloudFormation asume cuando opera con recursos en la pila especificada.

Output file name (Nombre de archivo de salida)

En la sección Advanced (Avanzados), puede especificar un nombre de archivo de salida, como `CreateStackOutput.json`, que CodePipeline añade al [artefacto de salida](#) después de realizar la acción especificada. El artefacto de salida contiene un archivo JSON con el contenido de la sección Outputs de la plantilla de CloudFormation.

Si no especifica un nombre, CodePipeline no genera un artefacto de salida.

Parameter overrides (Sobrescritura de parámetros)

Los parámetros se definen en la plantilla y le permiten ingresar valores personalizados al crear o actualizar una pila. Puede especificar un objeto JSON que anule los valores de los parámetros

de la plantilla en el archivo de configuración de la plantilla. Todos los nombres de parámetros deben estar presentes en la plantilla de pila. Para obtener más información, consulte [Sintaxis de Parameters de la plantilla de CloudFormation](#).

 Note

El límite máximo de tamaño del objeto JSON que se puede almacenar en la propiedad `ParameterOverrides` es de 1 kilobyte.

Le recomendamos que use el archivo de configuración de la plantilla para especificar la mayoría de sus valores de parámetros. Utilice anulaciones de parámetros para especificar solo valores de parámetros dinámicos. Los parámetros dinámicos son desconocidos hasta que se ejecuta la canalización.

En el siguiente ejemplo, se define un valor para el parámetro `ParameterName` usando una función de anulación de parámetros. La función recupera un valor de un artefacto de entrada de CodePipeline. Para obtener más información acerca de las funciones de anulación de parámetros, consulte [Uso de funciones de anulación de parámetros con canalizaciones de CodePipeline](#).

```
{
  "ParameterName" : { "Fn::GetParam" : ["ArtifactName", "config-file-name.json",
    "ParamName"]}
}
```

Propiedades de configuración (objeto JSON)

Cuando especifique CloudFormation como proveedor de la acción de una fase, defina las siguientes propiedades en la propiedad `Configuration`. Utilice el objeto JSON para las plantillas de la AWS CLI, la API de CodePipeline o CloudFormation. Para ver ejemplos, consulte [Tutorial: Creación de una canalización para pilas de prueba y pilas de producción y CloudFormationReferencia de propiedades de configuración de](#).

ActionMode

La acción de CloudFormation que CodePipeline invoca al procesar la fase asociada. Especifique solo uno de los siguientes modos de acción:

- `CHANGE_SET_EXECUTE` ejecuta un conjunto de cambios.
- `CHANGE_SET_REPLACE` crea el conjunto de cambios, si no existe, basándose en el nombre de la pila y la plantilla que envíe. Si el conjunto de cambios ya existe, CloudFormation lo elimina y luego crea otro nuevo.
- `CREATE_UPDATE` crea la pila si la pila especificado no existe. Si la pila existe, CloudFormation lo actualiza. Utilice esta acción para actualizar pilas existentes. CodePipeline no sustituirá la pila.
- `DELETE_ONLY` elimina una pila. Si especifica una pila que no existe, la acción se realiza de forma satisfactoria sin eliminar una pila.
- `REPLACE_ON_FAILURE` crea una pila si la pila especificada no existe. Si la pila existe y está en un estado de error (notificado como `ROLLBACK_COMPLETE`, `ROLLBACK_FAILED`, `CREATE_FAILED`, `DELETE_FAILED` o `UPDATE_ROLLBACK_FAILED`), CloudFormation elimina la pila y luego crea otra nueva. Si la pila no está en un estado de error, CloudFormation lo actualiza. Utilice esta acción para sustituir automáticamente pilas fallidas sin recuperarlos ni solucionar sus problemas. Por lo general, debería elegir este modo para las pruebas.

Esta propiedad es obligatoria.

Capabilities

Respecto a las pilas que contengan determinados recursos, reconozca explícitamente que CloudFormation podría crear o actualizar dichos recursos. Por ejemplo, debe especificar `CAPABILITY_IAM` si la plantilla de su pila contiene recursos de AWS Identity and Access Management (IAM). Para obtener más información, consulte los parámetros [CreateStack](#) de solicitud de operación de API.

Esta propiedad es condicional. Si dispone de recursos de IAM en su plantilla de pila, debe especificar esta propiedad.

Puede especificar varias capacidades. El siguiente ejemplo agrega las propiedades `CAPABILITY_IAM` y `CAPABILITY_AUTO_EXPAND` a la plantilla:

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

El nombre de un conjunto de cambios existentes o un nuevo conjunto de cambios que desee crear para la pila especificada.

Esta propiedad es necesaria para los siguientes modos de acción: `CHANGE_SET_REPLACE` y `CHANGE_SET_EXECUTE`. Para todos los demás modos de acción, se pasa por alto esta propiedad.

OutputFileName

Un nombre para el archivo de salida, como `CreateStackOutput.json`. CodePipeline agrega el archivo para el [artefacto de salida](#) después de llevar a cabo la acción especificada. El artefacto de salida contiene un archivo JSON con el contenido de la sección `Outputs` de la plantilla de CloudFormation.

Esta propiedad es opcional. Si no especifica un nombre, CodePipeline no genera un artefacto de salida.

ParameterOverrides

Los parámetros se definen en la plantilla y le permiten ingresar valores personalizados al crear o actualizar una pila. Puede especificar un objeto JSON que anule los valores de los parámetros de la plantilla en el archivo de configuración de la plantilla. Todos los nombres de parámetros deben estar presentes en la plantilla de pila. Para obtener más información, consulte [Sintaxis de Parameters de la plantilla de CloudFormation](#).

En el siguiente ejemplo se añaden las anulaciones de parámetros `InstanceType` y `KeyName` a la plantilla:

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

El tamaño máximo del objeto JSON que se puede almacenar en la propiedad `ParameterOverrides` es de 1 kilobyte.

Le recomendamos que use el archivo de configuración de la plantilla para especificar la mayoría de sus valores de parámetros. Utilice anulaciones de parámetros para especificar solo valores de parámetros dinámicos. Los valores de los parámetros dinámicos son desconocidos hasta que se ejecuta la canalización.

En el siguiente ejemplo, se define un valor para el parámetro `ParameterName` usando una función de anulación de parámetros. La función recupera un valor de un artefacto de entrada de CodePipeline. Para obtener más información acerca de las funciones de anulación de parámetros, consulte [Uso de funciones de anulación de parámetros con canalizaciones de CodePipeline](#).

```
{
  "ParameterName" : { "Fn::GetParam" : ["ArtifactName", "config-file-name.json",
    "ParamName"]}
}
```

Esta propiedad es opcional.

RoleArn

El nombre de recurso de Amazon (ARN) del rol de servicio de IAM que CloudFormation asume cuando opera con recursos en una pila.

Esta propiedad es necesaria para los siguientes modos de acción: `CREATE_UPDATE`, `REPLACE_ON_FAILURE`, `DELETE_ONLY` y `CHANGE_SET_REPLACE`. `RoleArn` no se aplica a la hora de ejecutar un conjunto de cambios. Si no utiliza CodePipeline para crear el conjunto de cambios, debe asegurarse de que el conjunto de cambios o la pila tenga un rol asociado.

StackName

El nombre de una pila existente o una pila que desee crear.

Esta propiedad es obligatoria para todos los demás modos de acción.

TemplateConfiguration

`TemplateConfiguration` es el archivo de configuración de la plantilla. Incluya el archivo en un artefacto de entrada en esta acción. El nombre del archivo de configuración de la plantilla se ajusta a este formato:

Artifactname::TemplateConfigurationFileName

`Artifactname` es el nombre del artefacto de entrada, tal y como aparece en CodePipeline. Por ejemplo, una etapa de origen con el nombre de artefacto de `SourceArtifact` y un nombre de archivo `test-configuration.json` crea un nombre `TemplateConfiguration`, tal y como se muestra en este ejemplo:

```
"TemplateConfiguration": "SourceArtifact::test-configuration.json"
```

El archivo de configuración de la plantilla puede contener los valores de los parámetros de la plantilla y una política de pila. Si incluye información confidencial, como contraseñas, restrinja el acceso a este archivo. Para ver un archivo de configuración de plantilla de ejemplo, consulte [CloudFormationArtefactos de](#).

Esta propiedad es opcional.

TemplatePath

TemplatePath representa el archivo de plantilla de CloudFormation. Incluya el archivo en un artefacto de entrada en esta acción. El nombre del archivo sigue este formato:

Artifactname::TemplateName

Artifactname es el nombre del artefacto de entrada, tal y como aparece en CodePipeline. Por ejemplo, una etapa de origen con el nombre de artefacto de SourceArtifact y un nombre de archivo template.yaml crea un nombre TemplatePath, tal y como se muestra en este ejemplo:

```
"TemplatePath": "SourceArtifact::template.yaml"
```

Esta propiedad es necesaria para los siguientes modos de acción: CREATE_UPDATE, REPLACE_ON_FAILURE y CHANGE_SET_REPLACE. Para todos los demás modos de acción, se pasa por alto esta propiedad.

Véase también

Los recursos relacionados siguientes pueden serle de ayuda cuando trabaje con estos parámetros.

- Para obtener más información sobre los parámetros de acción de CloudFormation en CodePipeline, consulte la [referencia de configuración de acciones de CloudFormation](#) en la Guía del usuario de AWS CodePipeline.
- Para ver valores de plantilla de ejemplo según el proveedor de acción, como para el campo Owner o los campos configuration, consulte la [Action structure reference](#) (Referencia de estructura de acciones) en la Guía del usuario de AWS CodePipeline.
- Para descargar las plantillas de la pila de canalización de ejemplo en formato YAML o JSON, consulte el [Tutorial: Creación de una canalización con CloudFormation](#) en la Guía del usuario de AWS CodePipeline.

CloudFormationArtefactos de

CodePipeline realiza tareas en artefactos mientras CodePipeline ejecuta una canalización. Para CloudFormation, los artefactos pueden incluir un archivo de plantilla de pila, un archivo de configuración de plantilla o ambos. CodePipeline utiliza estos artefactos para trabajar con pilas y conjuntos de cambios de CloudFormation.

Si utiliza Amazon Simple Storage Service (Amazon S3) como un repositorio de origen, deberá comprimir la plantilla y los archivos de configuración de la plantilla en un solo archivo antes de cargarlos en un bucket de S3. Para otros repositorios, como GitHub y AWS CodeCommit, cargue los artefactos sin comprimirlos. Para obtener más información, consulte [Crear una acción, etapas y acciones](#) en la Guía del usuario de AWS CodePipeline.

Puede añadir tantos archivos como necesite a su repositorio. Por ejemplo, es posible que desee incluir dos configuraciones diferentes para la misma plantilla: una para la configuración de prueba y otra para una configuración de producción.

Este tema describe cada tipo de artefacto.

Temas

- [Archivo de plantilla de pila](#)
- [Archivo de configuración de plantillas](#)
- [Véase también](#)

Archivo de plantilla de pila

Un archivo de plantilla de pila define los recursos que aprovisiona y configura CloudFormation. Estos archivos son los mismos archivos de plantillas que utiliza al crear o actualizar pilas mediante CloudFormation. Puede utilizar plantillas con formato YAML o JSON. Para obtener más información acerca de las plantillas, consulte [Secciones de la plantilla de CloudFormation](#).

Archivo de configuración de plantillas

Un archivo de configuración de plantillas es un archivo de texto con formato JSON que puede especificar los valores de los parámetros de plantilla, una [política de pila](#) y etiquetas. Utilice estos archivos de configuración para especificar valores de parámetros o una política para una pila. Todos los valores de los parámetros que especifique se deben declarar en la plantilla asociada.

Si incluye información confidencial, como contraseñas, en este archivo, restrinja el acceso a este archivo. Por ejemplo, si carga el artefacto en un bucket de S3, utilice [políticas de bucket de S3 o políticas de usuario](#) para restringir el acceso.

Para crear un archivo de configuración, utilice el siguiente formato:

```
{
  "Parameters" : {
    "NameOfTemplateParameter" : "ValueOfParameter",
    ...
  },
  "Tags" : {
    "TagKey" : "TagValue",
    ...
  },
  "StackPolicy" : {
    "Statement" : [
      StackPolicyStatement
    ]
  }
}
```

El siguiente ejemplo especifica TestEC2Key para el parámetro KeyName, añade una etiqueta Department cuyo valor es Marketing y añade una política de pila que permite todas las acciones de actualización, excepto una actualización que elimine un recurso.

```
{
  "Parameters" : {
    "KeyName" : "TestEC2Key"
  },
  "Tags" : {
    "Department" : "Marketing"
  },
  "StackPolicy" : {
    "Statement" : [
      {
        "Effect" : "Allow",
        "NotAction" : "Update:Delete",
        "Principal" : "*",
        "Resource" : "*"
      }
    ]
  }
}
```

```
}  
}
```

Véase también

Los recursos relacionados siguientes pueden serle de ayuda cuando trabaje con estos parámetros.

- Para obtener más información sobre los parámetros de acción de CloudFormation en CodePipeline, consulte la [referencia de configuración de acciones de CloudFormation](#) en la Guía del usuario de AWS CodePipeline.
- Para ver valores de plantilla de ejemplo según el proveedor de acción, como para el campo `Owner` o los campos `configuration`, consulte la [Action structure reference](#) (Referencia de estructura de acciones) en la Guía del usuario de AWS CodePipeline.
- Para descargar las plantillas de la pila de canalización de ejemplo en formato YAML o JSON, consulte el [Tutorial: Creación de una canalización con CloudFormation](#) en la Guía del usuario de AWS CodePipeline.

Uso de funciones de anulación de parámetros con canalizaciones de CodePipeline

En una fase de CodePipeline, puede especificar [anulaciones de parámetros](#) para acciones de CloudFormation. Las anulaciones de parámetros le permiten especificar valores de parámetros de pantalla que anulan los valores de un archivo de configuración de plantilla. CloudFormation proporciona funciones para ayudarle a especificar valores dinámicos (valores que son desconocidos hasta que se ejecuta la canalización).

Temas

- [Fn::GetArtifactAtt](#)
- [Fn::GetParam](#)
- [Véase también](#)

Fn::GetArtifactAtt

La función `Fn::GetArtifactAtt` recupera el valor de un atributo de un artefacto de entrada, como el nombre del bucket de S3 donde se almacena el artefacto. Utilice esta función para especificar los atributos de un artefacto, como su nombre de archivo o el nombre del bucket de Amazon S3.

Cuando ejecuta una canalización, CodePipeline copia y escribe los archivos en el almacén de artefactos de la canalización (un bucket de S3). CodePipeline genera los nombres de archivo en el almacén de artefactos. Estos nombres de archivos son desconocidos antes de ejecutar la canalización.

Por ejemplo, en la canalización, podría tener una fase de origen en la que CodePipeline copia el código fuente de su función de AWS Lambda en el almacén de artefactos. En la siguiente fase, dispone de una plantilla de CloudFormation que crea la función de Lambda, pero CloudFormation requiere el nombre de archivo para crear la función. Debe utilizar la función `Fn::GetArtifactAtt` para transferir los nombres de archivo y bucket de S3 exactos.

Sintaxis

Utilice la siguiente sintaxis para recuperar un valor de atributo de un artefacto.

```
{ "Fn::GetArtifactAtt" : [ "artifactName", "attributeName" ] }
```

`artifactName`

El nombre del artefacto de entrada. Debe declarar este artefacto como entrada para la acción asociada.

`attributeName`

El nombre del atributo de artefacto cuyo valor desea recuperar. Para más detalles acerca de cada atributo de artefacto, consulte la siguiente sección de atributos.

Ejemplo

Las siguientes anulaciones de parámetros especifican los parámetros `BucketName` y `ObjectKey` al recuperar el nombre del bucket de S3 y el nombre de archivo del artefacto `LambdaFunctionSource`. Este ejemplo presupone que CodePipeline copió el código fuente de la función de Lambda y lo guardó como un artefacto, por ejemplo, como parte de una fase de origen.

```
{  
  "BucketName" : { "Fn::GetArtifactAtt" : ["LambdaFunctionSource", "BucketName"] },  
  "ObjectKey" : { "Fn::GetArtifactAtt" : ["LambdaFunctionSource", "ObjectKey"]  
}
```


Atributos

Puede recuperar los siguientes atributos para un artefacto.

BucketName

El nombre del bucket de S3 donde se almacena el artefacto.

ObjectKey

El nombre del archivo .zip que contiene el artefacto que CodePipeline genera, como 1ABCyZZ.zip.

URL

La URL de Amazon Simple Storage Service (Amazon S3) del artefacto, como https://s3.us-west-2.amazonaws.com/artifactstorebucket-yivczw8jma0c/test/TemplateSo/1ABCyZZ.zip.

Fn::GetParam

La función Fn::GetParam devuelve un valor de un par clave-valor en un archivo con formato JSON. El archivo JSON debe estar incluido en un artefacto.

Utilice esta función para recuperar los valores de salida desde una pila de CloudFormation y usarlos como entrada para otra acción. Por ejemplo, si especifica un nombre de archivo de salida para una acción de CloudFormation, CodePipeline guarda la salida en un archivo JSON y luego lo agrega al archivo .zip del artefacto de salida. Utilice la función Fn::GetParam para recuperar el valor de salida y úselo como entrada para otra acción.

Sintaxis

Utilice la siguiente sintaxis para recuperar un valor de un par clave-valor.

```
{ "Fn::GetParam" : [ "artifactName", "JSONFileName", "keyName" ] }
```

artifactName

El nombre del artefacto, que debe incluirse como artefacto de entrada para la acción asociada.

JSONFileName

El nombre de un archivo JSON contenido en el artefacto.

keyName

El nombre de la clave cuyo valor desea recuperar.

Ejemplos

Los siguientes ejemplos muestran cómo utilizar la función `Fn::GetParam` en una anulación de parámetros.

Sintaxis

La siguiente anulación de parámetros especifica el parámetro `WebSiteURL` recuperando el valor de la clave `URL` del archivo `stack-output.json` que está en el artefacto `WebStackOutput`.

```
{
  "WebSiteURL" : { "Fn::GetParam" : ["WebStackOutput", "stack-output.json", "URL"]}
}
```

CloudFormationFragmentos de plantillas de

En los siguientes fragmentos de plantilla de CloudFormation, provenientes de una canalización de CodePipeline, se muestra cómo transferir salidas de la pila. Estos fragmentos muestran dos etapas de definición de la canalización. La primera etapa crea una pila y guarda sus salidas en el archivo `TestOutput.json` en el artefacto `StackAOutput`. Estos valores se especifican con las propiedades `OutputFileName` y `OutputArtifacts`.

El nombre del artefacto de entrada de origen para las etapas es `TemplateSource`. El nombre de archivo de la plantilla de pila es `teststackA.yaml` y el nombre de archivo del archivo de configuración es `test-configuration.json`. En ambas etapas, estos valores se especifican con las propiedades `TemplateConfiguration` y `TemplatePath`, como se muestra a continuación:

```
TemplateConfiguration: TemplateSource::test-configuration.json
TemplatePath: TemplateSource::teststackA.yaml
```

Example Etapa de creación de pila 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'

```

En una fase siguiente, la pila B utiliza las salidas de la pila A. En la propiedad `ParameterOverrides`, el ejemplo utiliza la función `Fn::GetParam` para especificar el parámetro `StackBInputParam`. El valor resultante es el valor asociado a la clave `StackAOutputName`.

Example Etapa de creación de pila 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" : ["StackAOutput", "TestOutput.json",
"StackAOutputName"]}

```

```
}  
InputArtifacts:  
  - Name: TemplateSource  
  - Name: StackAOutput  
RunOrder: '1'
```

Véase también

Los recursos relacionados siguientes pueden serle de ayuda cuando trabaje con estos parámetros.

- Para obtener más información sobre los parámetros de acción de CloudFormation en CodePipeline, consulte la [referencia de configuración de acciones de CloudFormation](#) en la Guía del usuario de AWS CodePipeline.
- Para ver valores de plantilla de ejemplo según el proveedor de acción, como para el campo `Owner` o los campos `configuration`, consulte la [Action structure reference](#) (Referencia de estructura de acciones) en la Guía del usuario de AWS CodePipeline.
- Para descargar las plantillas de la pila de canalización de ejemplo en formato YAML o JSON, consulte el [Tutorial: Creación de una canalización con CloudFormation](#) en la Guía del usuario de AWS CodePipeline.

La guía de referencia de plantillas de CloudFormation

Ahora puede encontrar información detallada sobre los recursos y componentes disponibles para crear plantillas de CloudFormation en la [Guía de referencia de plantillas de CloudFormation](#). Esta guía sirve como referencia técnica para las plantillas y complementa el contenido de introducción de la Guía del usuario de CloudFormation.

La Guía de referencia de plantillas de CloudFormation proporciona detalles sobre lo siguiente:

- Tipos de recursos y sus propiedades
- DeletionPolicy y otros atributos a nivel de recurso
- Funciones intrínsecas, como Ref y Sub
- Transformaciones (macros de CloudFormation)
- Scripts auxiliares para la configuración de instancias de EC2
- Definiciones de tipos de recursos legibles por una máquina

Esta guía se actualiza periódicamente para reflejar lo siguiente:

- Nuevos tipos de recursos
- Cambios de propiedades
- Nuevos valores de retorno
- Actualizaciones de la documentación

Seguridad en CloudFormation

La seguridad en AWS es la principal prioridad. Como cliente de AWS, se beneficiará de una arquitectura de red y de centros de datos diseñados para satisfacer los requisitos de seguridad de las organizaciones más exigentes.

La seguridad es una responsabilidad compartida entre AWS y usted. El [modelo de responsabilidad compartida](#) la describe como seguridad de la nube y seguridad en la nube:

- Seguridad de la nube: AWS es responsable de proteger la infraestructura que ejecuta los servicios de AWS en la nube de AWS. AWS también proporciona servicios que puede utilizar de forma segura. Los auditores externos prueban y verifican periódicamente la eficacia de nuestra seguridad como parte de los [AWS Programas de conformidad de](#) . Para obtener información sobre los programas de conformidad que se aplican a CloudFormation, consulte [AWS Services in Scope by Compliance Program](#).
- Seguridad en la nube: su responsabilidad se determina según el servicio de AWS que utiliza. También es responsable de otros factores, incluida la confidencialidad de los datos, los requisitos de la empresa y la legislación y la normativa aplicables.

Esta documentación le ayuda a comprender cómo aplicar el modelo de responsabilidad compartida a la hora de utilizar CloudFormation. Los siguientes temas le mostrarán cómo configurar CloudFormation para satisfacer sus objetivos de seguridad y de conformidad. También puede aprender a utilizar otros servicios de AWS que lo ayuden a monitorizar y proteger los recursos de CloudFormation.

Temas

- [Protección de las pilas de CloudFormation para evitar la eliminación](#)
- [Impida actualizaciones en los recursos de la pila](#)
- [Protección de los datos en CloudFormation](#)
- [Control del acceso a CloudFormation con AWS Identity and Access Management](#)
- [Registro de llamadas a la API de CloudFormation con AWS CloudTrail](#)
- [Seguridad de la infraestructura en \(\) CloudFormation](#)
- [Resiliencia en CloudFormation](#)
- [Validación de conformidad en CloudFormation](#)
- [Configuración y análisis de vulnerabilidades en CloudFormation](#)

- [Prácticas recomendadas de seguridad para CloudFormation](#)
- [Acceso a CloudFormation a través de un punto de conexión de la interfaz \(AWS PrivateLink\)](#)

Protección de las pilas de CloudFormation para evitar la eliminación

Es posible evitar que una pila se elimine de manera accidental; para ello, habilite la protección de terminación en la pila. Si un usuario intenta eliminar una pila con la protección de terminación habilitada, la eliminación fallará y la pila junto con su estado no cambiarán. Es posible habilitar la protección de terminación en una pila al crearla. La protección de terminación no está habilitada de manera predeterminada en las pilas. Puede configurar la protección de terminación en una pila con cualquier estado excepto `DELETE_IN_PROGRESS` o `DELETE_COMPLETE`.

Al habilitar o deshabilitar la protección de terminación en una pila, también se habilita o deshabilita en las pilas anidadas. No se puede habilitar o deshabilitar la protección de terminación directamente en una pila anidada. Si un usuario intenta eliminar directamente una pila anidada que pertenece a una pila la con protección de terminación habilitada, la operación fallará y la pila anidada no cambiará.

Sin embargo, si un usuario realiza una actualización de la pila que eliminaría la pila anidada, CloudFormation la elimina.

La protección de terminación no es igual a la deshabilitación de restauración. La protección de terminación solo es posible en los intentos de eliminar pilas, mientras que la deshabilitación de restauración es posible en la restauración automática cuando se produce algún error durante la creación de la pila.

Habilitación de la protección de terminación al crear una pila

En la página Specify stack options (Especificar opciones de pila) del asistente Create stack (Crear pila), en Advanced options (Opciones avanzadas), amplíe la sección Termination Protection (Protección de terminación) y seleccione Enable (Habilitar). Para obtener más información, consulte [Configurar las opciones la pila](#).

Habilitación o deshabilitación de la protección de terminación de una pila

1. Inicie sesión en la Consola de administración de AWS y abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation>.
2. En la barra de navegación de la parte superior de la pantalla, elija la Región de AWS.

3. Seleccione la pila que desee.

Note

Si aparece el texto NESTED junto al nombre de la pila, es una pila anidada. Solo es posible cambiar la protección de terminación de la pila raíz a la que pertenece la pila anidada.

4. En el panel de detalles de la pila, seleccione Stack actions (Acciones de la pila) y, a continuación, haga clic en Edit termination protection (Editar protección de terminación).

CloudFormation muestra el cuadro de diálogo Editar protección de terminación.

5. Elija Enable (Habilitar) o Disable (Deshabilitar) y, a continuación, seleccione Save (Guardar).

Habilitación o deshabilitación de la protección de terminación de una pila anidada

Si aparece el texto NESTED junto al nombre de la pila, es una pila anidada. Solo es posible cambiar la protección de terminación de la pila raíz a la que pertenece la pila anidada. Cambio de la protección de terminación de la pila raíz:

1. Inicie sesión en la Consola de administración de AWS y abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation>.
2. En la barra de navegación de la parte superior de la pantalla, elija la Región de AWS.
3. Seleccione la pila anidada que desee.
4. En el panel Stack info (Información de la pila), en la sección Overview (Información general), seleccione el nombre de la pila mostrada como Root stack (Pila raíz).

CloudFormation muestra los detalles de la pila raíz.

5. Seleccione Stack actions (Acciones de la pila) y, a continuación, seleccione Edit Termination Protection (Editar protección de terminación).

CloudFormation muestra el cuadro de diálogo Editar protección de terminación.

6. Elija Enable (Habilitar) o Disable (Deshabilitar) y, a continuación, seleccione Save (Guardar).

Habilitación o deshabilitación de la protección de terminación con la línea de comandos

Utilice el comando [update-termination-protection](#).

Control de quién puede cambiar la protección de terminación de pilas

Para habilitar o deshabilitar la protección de terminación de pilas, un usuario requiere permiso para la acción `cloudformation:UpdateTerminationProtection`. Por ejemplo, la política que figura a continuación permite a los usuarios habilitar o deshabilitar la protección de terminación de pilas.

Para obtener más información acerca de cómo especificar permisos en CloudFormation, consulte [Control del acceso a CloudFormation con AWS Identity and Access Management](#).

Example Un ejemplo de política que concede permisos para cambiar la protección de terminación de pilas

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "cloudformation:UpdateTerminationProtection"
    ],
    "Resource": "*"
  }]
}
```

Impida actualizaciones en los recursos de la pila

Al crear una pila, todas las acciones de actualización están permitidas en todos los recursos. De forma predeterminada, cualquiera con permisos de actualización de pilas puede actualizar todos los recursos de la pila. Durante una actualización, algunos recursos podría necesitar una interrupción o ser sustituidos por completo, lo que se traduce en nuevos ID físicos o almacenamiento completamente nuevo. Puede impedir la actualización o eliminación involuntaria de recursos de pila durante la actualización de una pila mediante una política de pilas. Una política de pilas es un documento de JSON que describe las acciones de actualización que pueden realizarse en recursos designados.

Después de establecer una política de pilas, todos los recursos de la pila están protegidos de forma predeterminada. Para permitir actualizaciones en recursos concretos, usted especifica una

declaración Allow explícita para esos recursos en su política de pilas. Solo es posible definir una política de pilas por pila, pero puede proteger varios recursos dentro de una sola política. Una política de pilas se aplica a todos los usuarios de CloudFormation que traten de actualizar la pila. No es posible asociar diferentes políticas de pilas a diferentes usuarios.

La política de pilas se aplica únicamente durante las actualizaciones de la pila. No ofrece controles de acceso como una política de AWS Identity and Access Management (IAM). Utilice una política de pilas solo como un mecanismo a prueba de errores para evitar actualizaciones accidentales de determinados recursos de pila. Para controlar el acceso a recursos o acciones de AWS, utilice IAM.

Temas

- [Definición de una política de pilas](#)
- [Definición de una política de pilas](#)
- [Establecimiento de una política de pilas](#)
- [Actualización de recursos protegidos](#)
- [Modificación de una política de pilas](#)
- [Más políticas de pilas de ejemplo](#)

Definición de una política de pilas

El siguiente ejemplo de política de pilas impide actualizaciones al recurso ProductionDatabase:

```
{
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "*"
    },
    {
      "Effect" : "Deny",
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "LogicalResourceId/ProductionDatabase"
    }
  ]
}
```

Cuando establece una política de pilas, todos los recursos de la pila están protegidos de forma predeterminada. Para permitir la actualización de todos los recursos, añadimos una declaración `Allow` que permite todas las acciones en todos los recursos. Aunque la declaración `Allow` especifica todos los recursos, la declaración `Deny` explícita invalida el permiso para el recurso con el ID lógico `ProductionDatabase`. Esta declaración `Deny` impide todas las acciones de actualización, como, por ejemplo, la sustitución o eliminación, en el recurso `ProductionDatabase`.

El elemento `Principal` es necesario, pero solo admite el comodín (`*`), lo que significa que la declaración se aplica a todas las [entidades principales](#).

Note

Durante una actualización de la pila, CloudFormation actualiza automáticamente los recursos que dependen de otros recursos actualizados. Por ejemplo, CloudFormation actualiza un recurso que haga referencia a un recurso actualizado. CloudFormation no realiza cambios físicos, como el ID de recursos, a recursos que se han actualizado automáticamente, pero si hay una política de pilas asociada a esos recursos, debe tener permiso para actualizarlos.

Definición de una política de pilas

Al crear una pila, no hay ninguna política de pilas establecida, por lo que todas las acciones de actualización están permitidas en todos los recursos. Para proteger los recursos de la pila de acciones de actualización, defina una política de pilas y, a continuación, establézcala en la pila. Una política de pilas es un documento JSON que define las acciones de actualización de la pila de CloudFormation que los usuarios de CloudFormation pueden realizar y los recursos a los que se aplican las acciones. Usted establece la política de pilas al crear una pila, especificando un archivo de texto que contiene la política de pilas o escribiéndola. Al establecer una política de pilas en la pila, por defecto se deniega cualquier actualización que no esté permitida de forma explícita.

Puede definir una política de pilas con cinco elementos: `Effect`, `Action`, `Principal`, `Resource` y `Condition`. El siguiente pseudocódigo muestra la sintaxis de la política de la pila.

```
{
  "Statement" : [
    {
      "Effect" : "Deny_or_Allow",
      "Action" : "update_actions",
      "Principal" : "*",
```

```
"Resource" : "LogicalResourceId/resource_logical_ID",
"Condition" : {
  "StringEquals_or_StringLike" : {
    "ResourceType" : [resource_type, ...]
  }
}
]
```

Effect

Determina si las acciones que especifique se denegarán o permitirán en el recurso o recursos que especifique. Puede especificar solamente Deny o Allow, como por ejemplo:

```
"Effect" : "Deny"
```

Important

Si una política de pilas incluye declaraciones que se solapan (permitiendo y denegando actualizaciones en un recurso), una declaración Deny siempre anula una declaración Allow. Para garantizar que un recurso está protegido, utilice una declaración Deny para dicho recurso.

Action

Especifica las acciones de actualización denegadas o permitidas:

Update:Modify

Especifica acciones de actualización durante las cuales los recursos podrían no experimentar interrupciones o algunas interrupciones mientras se aplican cambios. Todos los recursos mantienen sus ID físicos.

Update:Replace

Especifica acciones de actualización durante las cuales se vuelven a crear recursos. CloudFormation crea un nuevo recurso con las actualizaciones especificadas y, a continuación, elimina el recurso anterior. Dado que se ha vuelto a crear el recurso, el ID físico del recurso nuevo podría ser diferente.

Update:Delete

Especifica acciones de actualización durante las cuales se eliminan recursos. Las actualizaciones que eliminan recursos completamente de una plantilla de pila requieren esta acción.

Update:*

Especifica todas las acciones de actualización. El asterisco es un comodín que representa todas las acciones de actualización.

El siguiente ejemplo muestra cómo especificar tan solo las acciones de sustitución y eliminación:

```
"Action" : ["Update:Replace", "Update:Delete"]
```

Para permitir todas las acciones de actualización excepto una, utilice `NotAction`. Por ejemplo, para permitir todas las acciones de actualización excepto `Update:Delete`, utilice `NotAction`, tal y como se muestra en este ejemplo:

```
{
  "Statement" : [
    {
      "Effect" : "Allow",
      "NotAction" : "Update:Delete",
      "Principal": "*",
      "Resource" : "*"
    }
  ]
}
```

Principal

El elemento `Principal` especifica la entidad a la que se aplica la política. El elemento es necesario, pero solo admite el comodín (*), lo que significa que la política se aplica a todas las [entidades principales](#).

Recurso

Especifica el ID lógico de los recursos a los que se aplica a la política. Para especificar tipos de recursos, utilice el elemento `Condition`.

Para especificar un único recurso, utilice su ID lógico. Por ejemplo:

```
"Resource" : ["LogicalResourceId/myEC2instance"]
```

Puede utilizar un comodín con ID lógicos. Por ejemplo, si utiliza un prefijo de ID lógico común para todos los recursos relacionados, puede especificar todos ellos con un comodín:

```
"Resource" : ["LogicalResourceId/CriticalResource*"]
```

También puede utilizar un elemento Not con recursos. Por ejemplo, para permitir actualizaciones de todos los recursos a excepción de uno, utilice un elemento NotResource para proteger dicho recurso:

```
{
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : "Update:*",
      "Principal": "*",
      "NotResource" : "LogicalResourceId/ProductionDatabase"
    }
  ]
}
```

Al establecer una política de pilas, se deniega cualquier actualización que no esté permitida de forma explícita. Al permitir actualizaciones de todos los recursos, excepto para el recurso ProductionDatabase, deniega las actualizaciones del recurso ProductionDatabase.

Condiciones

Especifica el tipo de recurso al que se aplica la política. Para especificar el ID lógico de recursos concretos, utilice el elemento Resource.

Puede especificar un tipo de recurso, como todas las instancias de base de datos de RDS y EC2, tal y como se muestra en el siguiente ejemplo:

```
{
  "Statement" : [
    {
      "Effect" : "Deny",
      "Principal" : "*",
      "Action" : "Update:*",
      "Resource" : "*",
    }
  ]
}
```

```
"Condition" : {
  "StringEquals" : {
    "ResourceType" : ["AWS::EC2::Instance", "AWS::RDS::DBInstance"]
  }
},
{
  "Effect" : "Allow",
  "Principal" : "*",
  "Action" : "Update:*",
  "Resource" : "*"
}
```

La declaración `Allow` concede permisos de actualización para todos los recursos y la declaración `Deny` deniega actualizaciones a instancias de bases de datos de RDS y EC2. La declaración `Deny` siempre anula acciones de permiso.

Puede utilizar un comodín con tipos de recursos. Por ejemplo, puede denegar permisos de actualización para todos los recursos de Amazon EC2 (como, por ejemplo, instancias, grupos de seguridad y subredes) utilizando un comodín, tal y como se muestra en el siguiente ejemplo:

```
"Condition" : {
  "StringLike" : {
    "ResourceType" : ["AWS::EC2::*"]
  }
}
```

Debe utilizar la condición `StringLike` cuando utiliza comodines.

Establecimiento de una política de pilas

Puede utilizar la consola o la AWS CLI para aplicar una política de pilas al crear una pila. También puede utilizar la AWS CLI para aplicar una política de pilas a una pila existente. Después de aplicar una política de pilas, no puede eliminarla de la pila, pero sí puede utilizar la AWS CLI para modificarla.

Las políticas de pilas se aplican a todos los usuarios de CloudFormation que traten de actualizar la pila. No es posible asociar diferentes políticas de pilas a diferentes usuarios.

Para obtener información sobre la escritura de políticas de pilas, consulte [Definición de una política de pilas](#).

Para definir una política de pilas al crear una pila (consola)

1. Abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation>.
2. En la barra de navegación de la parte superior de la pantalla, elija la Región de AWS en la que creó la pila.
3. En la página CloudFormation Stacks (Pilas de CloudFormation), elija Create Stack (Crear pila).
4. En el asistente para crear pilas, en la página Configure stack options (Configurar opciones de pila), expanda la sección Advanced (Avanzado) y, a continuación, elija Stack policy (Política de pila).
5. Especifique la política de pila:
 - Para escribir una política directamente en la consola, elija Enter stack policy (Introducir política de pilas) y, a continuación, escriba la política de pilas directamente en el campo de texto.
 - Para utilizar una política definida en un archivo aparte, elija Upload a file (Cargar un archivo) y, a continuación, Choose file (Elegir archivo) para seleccionar el archivo que contiene la política de pilas.

Para definir una política de pilas al crear una pila (AWS CLI)

- Utilice el comando [create-stack](#) con la opción `--stack-policy-body` para escribir una política modificada o la opción `--stack-policy-url` para especificar un archivo que contenga la política.

Para establecer una política de pilas en una pila existente (AWS CLI solamente)

- Utilice el comando [set-stack-policy](#) con la opción `--stack-policy-body` para escribir una política modificada o la opción `--stack-policy-url` para especificar un archivo que contenga la política.

 Note

Para añadir una política a una pila existente, debe tener permiso para la acción [SetStackPolicy](#) de CloudFormation.

Actualización de recursos protegidos

Para actualizar recursos protegidos, cree una política temporal que anule la política de pilas y permita actualizaciones en esos recursos. Especifique la política de invalidación al actualizar la pila. La política de invalidación no cambia de manera permanente la política de pilas.

Para actualizar recursos protegidos, debe disponer de permiso para utilizar la acción [SetStackPolicy](#) de CloudFormation. Para obtener más información sobre el establecimiento de permisos de CloudFormation, consulte [Control del acceso a CloudFormation con AWS Identity and Access Management](#).

Note

Durante una actualización de la pila, CloudFormation actualiza automáticamente los recursos que dependen de otros recursos actualizados. Por ejemplo, CloudFormation actualiza un recurso que haga referencia a un recurso actualizado. CloudFormation no realiza cambios físicos, como el ID de recursos, a recursos que se han actualizado automáticamente, pero si hay una política de pilas asociada a esos recursos, debe tener permiso para actualizarlos.

Para actualizar un recurso protegido (consola)

1. Abra la consola de CloudFormation en <https://console.aws.amazon.com/cloudformation>.
2. Seleccione la pila que desea actualizar, elija Actions (Acciones) y, a continuación, elija Update Stack (Actualizar pila).
3. Si no ha modificado la plantilla de pilas, seleccione Use current template (Usar plantilla actual) y, a continuación, haga clic en Next (Siguiendo). Si ha modificado la plantilla, seleccione Replace current template (Sustituir plantilla actual) y especifique la ubicación de la plantilla actualizada en la sección Specify template (Especificar plantilla):
 - Para una plantilla almacenada localmente en su equipo, seleccione Upload a template file (Cargar un archivo de plantilla). Elija Choose File (Elegir archivo) para navegar hasta el archivo y seleccionarlo; a continuación, haga clic en Next (Siguiendo).
 - Para una plantilla almacenada en un bucket de Amazon S3, seleccione Amazon S3 URL (URL de Amazon S3). Escriba o pegue la URL para la plantilla y, a continuación, haga clic en Next (Siguiendo).

Si tiene una plantilla en un bucket compatible con el control de versiones, puede especificar una versión concreta de la plantilla al adjuntar `?versionId=version-id` a la URL. Para obtener más información, consulte [Trabajo con objetos en un bucket con control de versiones habilitado](#) en la Guía del usuario de la consola de Amazon Simple Storage Service.

4. Si la plantilla contiene parámetros, en la página Specify stack details (Especificar detalles de pila) escriba o modifique los valores de los parámetros y, a continuación, elija Next (Siguiente).

CloudFormation rellenará cada parámetro con el valor que actualmente está establecido en la pila excepto los parámetros declarados con el atributo NoEcho. Puede utilizar los valores actuales para los parámetros eligiendo Use existing value (Usar valor existente).

Para obtener más información sobre el uso de NoEcho para enmascarar información confidencial y el uso de los parámetros dinámicos para la administración de secretos, consulte la práctica recomendada de [No integre credenciales en sus plantillas](#).

5. Especifique una política de pila de sustitución.
 - a. En la página Configure stack options (Configurar opciones de pila), en la sección Advanced options (Opciones avanzadas), seleccione Stack policy (Política de pila).
 - b. Seleccione Upload a file (Cargar un archivo).
 - c. Haga clic en Choose file (Seleccionar un archivo) y desplácese hasta el archivo que contiene la política de pila de sustitución o escriba una política.
 - d. Elija Siguiente.

La política de invalidación debe especificar una declaración Allow para los recursos protegidos que desea actualizar. Por ejemplo, para actualizar todos los recursos protegidos, especifique una política de invalidación temporal que permita todas las actualizaciones:

```
{
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "*"
    }
  ]
}
```

 Note

CloudFormation aplica la política de invalidación solo durante esta actualización. La política de invalidación no cambia de manera permanente la política de pilas. Para modificar una política de pilas, consulte [Modificación de una política de pilas](#).

6. Revise la información de la pila y los cambios que ha enviado.

Compruebe que ha enviado la información correcta, como la URL de la plantilla o los valores de los parámetros correctos. Si la plantilla contiene recursos de IAM, elija `I acknowledge that this template may create IAM resources` (Confirmo que esta plantilla pueda crear recursos de IAM) para especificar que desea utilizar recursos de IAM en la plantilla. Para obtener más información, consulte [Reconocimiento de recursos de IAM en plantillas de CloudFormation](#).

En la sección Vista previa de los cambios, compruebe que CloudFormation realice todos los cambios que espera. Por ejemplo, compruebe que CloudFormation añada, elimine y modifique los recursos que usted pretendía añadir, eliminar o modificar. CloudFormation genera esta vista previa mediante la creación de un conjunto de cambios para la pila. Para obtener más información, consulte [Actualización de pilas de CloudFormation con conjuntos de cambios](#).

7. Cuando esté satisfecho con los cambios, haga clic en Update (Actualizar).

 Note

En este momento, también tiene la opción de ver el conjunto de cambios para revisar las actualizaciones propuestas más detenidamente. Para hacerlo, haga clic en View change set (Ver el conjunto de cambios) en lugar de Update (Actualizar). CloudFormation muestra el conjunto de cambios generados en función de sus actualizaciones. Cuando esté listo para actualizar la pila, haga clic en Execute (Ejecutar).

CloudFormation muestra la página Stack details (Detalles de pila) de la pila. La pila ahora tiene un estado `UPDATE_IN_PROGRESS`. Una vez que CloudFormation haya terminado correctamente de actualizar la pila, establece el estado de la pila en `UPDATE_COMPLETE`.

Si se produce un error en la actualización de la pila, CloudFormation restaura automáticamente los cambios y establece el estado de la pila en `UPDATE_ROLLBACK_COMPLETE`.

Para actualizar un recurso protegido (AWS CLI)

- Utilice el comando [update-stack](#) con la opción `--stack-policy-during-update-body` para escribir una política modificada o la opción `--stack-policy-during-update-url` para especificar un archivo que contenga la política.

Note

CloudFormation aplica la política de invalidación solo durante esta actualización. La política de invalidación no cambia de manera permanente la política de pilas. Para modificar una política de pilas, consulte [Modificación de una política de pilas](#).

Modificación de una política de pilas

Para proteger recursos adicionales o para eliminar la protección de recursos, modifique la política de pilas. Por ejemplo, cuando añada a su pila una base de datos que desea proteger, añada una declaración Deny para esa base de datos a la política de pilas. Para modificar la política, debe disponer de permiso para utilizar la acción [SetStackPolicy](#).

Utilice la AWS CLI para modificar las políticas de pilas.

Para modificar una política de pilas (AWS CLI)

- Utilice el comando [set-stack-policy](#) con la opción `--stack-policy-body` para escribir una política modificada o la opción `--stack-policy-url` para especificar un archivo que contenga la política.

No puede eliminar una política de pilas. Para eliminar toda la protección de todos los recursos, usted modifica la política para permitir explícitamente todas las acciones en todos los recursos. La siguiente política permite todas las actualizaciones en todos los recursos:

```
{
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "*"
    }
  ]
}
```

```
}  
]  
}
```

Más políticas de pilas de ejemplo

Las siguientes políticas de ejemplo muestran cómo evitar actualizaciones en todos los recursos de la pila y en recursos concretos, y cómo prevenir determinados tipos de actualizaciones.

Impida actualizaciones en todos los recursos de la pila

Para impedir las actualizaciones de todos los recursos de la pila, la siguiente política especifica una declaración Deny para todas las acciones de actualización en todos los recursos.

```
{  
  "Statement" : [  
    {  
      "Effect" : "Deny",  
      "Action" : "Update:*",  
      "Principal": "*",  
      "Resource" : "*"   
    }  
  ]  
}
```

Impida actualizaciones en un solo recurso

La siguiente política deniega todas las acciones de actualización en la base de datos con el ID lógico MyDatabase. Permite todas las acciones de actualización en los recursos de pila restantes con una declaración Allow. La declaración Allow no se aplica al recurso MyDatabase porque la declaración Deny siempre invalida las acciones de permiso.

```
{  
  "Statement" : [  
    {  
      "Effect" : "Deny",  
      "Action" : "Update:*",  
      "Principal": "*",  
      "Resource" : "LogicalResourceId/MyDatabase"   
    },  
    {
```

```
    "Effect" : "Allow",
    "Action" : "Update:*",
    "Principal": "*",
    "Resource" : "*"
  }
]
```

Puede conseguir el mismo resultado que el ejemplo anterior mediante una denegación predeterminada. Al establecer una política de pilas, CloudFormation deniega cualquier actualización que no esté permitida de forma explícita. La siguiente política permite las actualizaciones de todos los recursos, excepto para el recurso `ProductionDatabase`, que se deniega de forma predeterminada.

```
{
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : "Update:*",
      "Principal": "*",
      "NotResource" : "LogicalResourceId/ProductionDatabase"
    }
  ]
}
```

Important

Existe riesgo en la utilización de una denegación predeterminada. Si dispone de una declaración `Allow` en otro punto de la política (como, por ejemplo, una declaración `Allow` que utiliza un comodín), podría conceder sin saberlo permiso de actualización a recursos que no tenía la intención de hacerlo. Dado que una denegación explícita invalida cualquier acción de permiso, puede garantizar la protección de un recurso mediante una declaración `Deny`.

Impida las actualizaciones de todas las instancias de un tipo de recurso

La siguiente política deniega todas las acciones de actualización en el tipo de recurso de instancia de base de datos de RDS. Permite todas las acciones de actualización en los recursos de pila restantes con una declaración `Allow`. La declaración `Allow` no se aplica a los recursos de instancia de base de datos de RDS porque una declaración `Deny` siempre invalida las acciones de permiso.

```
{
  "Statement" : [
    {
      "Effect" : "Deny",
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "*",
      "Condition" : {
        "StringEquals" : {
          "ResourceType" : ["AWS::RDS::DBInstance"]
        }
      }
    },
    {
      "Effect" : "Allow",
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "*"
    }
  ]
}
```

Evite las actualizaciones de sustitución de una instancia

La siguiente política deniega las actualizaciones que provocarían una sustitución de la instancia por el ID lógico MyInstance. Permite todas las acciones de actualización en los recursos de pila restantes con una declaración Allow. La declaración Allow no se aplica al recurso MyInstance porque la declaración Deny siempre invalida las acciones de permiso.

```
{
  "Statement" : [
    {
      "Effect" : "Deny",
      "Action" : "Update:Replace",
      "Principal": "*",
      "Resource" : "LogicalResourceId/MyInstance"
    },
    {
      "Effect" : "Allow",
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "*"
    }
  ]
}
```

```
}  
]  
}
```

Evite las actualizaciones de pilas anidadas

La siguiente política deniega todas las acciones de actualización en el tipo de recurso de pila de CloudFormation (pilas anidadas). Permite todas las acciones de actualización en los recursos de pila restantes con una declaración Allow. La declaración Allow no se aplica a los recursos de pila de CloudFormation porque la declaración Deny siempre invalida las acciones de permiso.

```
{  
  "Statement" : [  
    {  
      "Effect" : "Deny",  
      "Action" : "Update:*",  
      "Principal" : "*",  
      "Resource" : "*",  
      "Condition" : {  
        "StringEquals" : {  
          "ResourceType" : ["AWS::CloudFormation::Stack"]  
        }  
      }  
    },  
    {  
      "Effect" : "Allow",  
      "Action" : "Update:*",  
      "Principal" : "*",  
      "Resource" : "*"   
    }  
  ]  
}
```

Protección de los datos en CloudFormation

El [modelo de responsabilidad compartida](#) de AWS se aplica a la protección de datos de AWS CloudFormation. Como se describe en este modelo, AWS es responsable de proteger la infraestructura global que ejecuta toda la Nube de AWS. Eres responsable de mantener el control sobre el contenido alojado en esta infraestructura. También eres responsable de las tareas de administración y configuración de seguridad para los Servicios de AWS que utiliza. Para obtener más

información sobre la privacidad de los datos, consulta las [Preguntas frecuentes sobre la privacidad de datos](#). Para obtener información sobre la protección de datos en Europa, consulta la publicación de blog sobre el [Modelo de responsabilidad compartida de AWS y GDPR](#) en el Blog de seguridad de AWS.

Con fines de protección de datos, recomendamos proteger las credenciales de la Cuenta de AWS y configurar cuentas de usuario individuales con AWS IAM Identity Center o AWS Identity and Access Management (IAM). De esta manera, solo se otorgan a cada usuario los permisos necesarios para cumplir sus obligaciones laborales. También recomendamos proteger sus datos de la siguiente manera:

- Utiliza la autenticación multifactor (MFA) en cada cuenta.
- Utiliza SSL/TLS para comunicarse con los recursos de AWS. Se recomienda el uso de TLS 1.2 y recomendamos TLS 1.3.
- Configure los registros de API y de actividad de los usuarios con AWS CloudTrail. Para obtener información sobre cómo utilizar registros de seguimiento de CloudTrail para capturar actividades de AWS, consulta [Working with CloudTrail trails](#) en la Guía del usuario de AWS CloudTrail.
- Utiliza las soluciones de cifrado de AWS, junto con todos los controles de seguridad predeterminados dentro de los servicios de Servicios de AWS.
- Utiliza servicios de seguridad administrados avanzados, como Amazon Macie, que lo ayuden a detectar y proteger los datos confidenciales almacenados en Amazon S3.
- Si necesita módulos criptográficos validados FIPS 140-3 al acceder a AWS a través de una interfaz de línea de comandos o una API, utiliza un punto de conexión de FIPS. Para obtener más información sobre los puntos de conexión de FIPS disponibles, consulta [Estándar de procesamiento de la información federal \(FIPS\) 140-3](#).

Se recomienda encarecidamente no introducir nunca información confidencial o sensible, como por ejemplo, direcciones de correo electrónico de clientes, en etiquetas o campos de formato libre, tales como el campo Nombre. Esto incluye las situaciones en las que debe trabajar con la CloudFormation u otros Servicios de AWS a través de la consola, la API, la AWS CLI o los SDK de AWS. Cualquier dato que ingrese en etiquetas o campos de texto de formato libre utilizados para nombres se puede emplear para los registros de facturación o diagnóstico. Si proporciona una URL a un servidor externo, recomendamos encarecidamente que no incluya la información de las credenciales en la URL para validar la solicitud para ese servidor.

Cifrado en reposo

En consonancia con el modelo de responsabilidad compartida de AWS, CloudFormation almacena los datos cifrados en reposo. Los clientes son responsables de establecer políticas de cifrado y almacenamiento para los datos almacenados en las cuentas. Por ejemplo, se recomienda habilitar el cifrado en reposo para plantillas y otros datos almacenados en buckets de S3 o temas de SNS. Los clientes definen de manera similar la configuración de cifrado para cualquier sistema de almacenamiento de datos aprovisionado por CloudFormation.

Cifrado en tránsito

CloudFormation utiliza canales cifrados para las comunicaciones de servicios en el modelo de responsabilidad compartida.

Privacidad del tráfico entre redes

Las comunicaciones de servicio de CloudFormation se cifran de forma segura de forma predeterminada entre regiones o zonas de disponibilidad.

Control del acceso a CloudFormation con AWS Identity and Access Management

Con AWS Identity and Access Management (IAM), puede crear usuarios de IAM para controlar quién tiene acceso a recursos específicos en su Cuenta de AWS. Cuando utiliza IAM, puede controlar lo que los usuarios pueden hacer con CloudFormation; por ejemplo, si pueden ver las plantillas de pila, crear pilas o eliminar pilas.

Además de acciones específicas de CloudFormation, puede administrar qué servicios y recursos de AWS están disponibles para cada usuario. De esa manera, puede controlar a qué recursos pueden obtener acceso los usuarios cuando utilizan CloudFormation. Por ejemplo, puede especificar qué usuarios pueden crear instancias Amazon EC2, terminar instancias de base de datos o actualizar VPC. Estos mismos permisos se aplican siempre que utilizan CloudFormation para realizar esas acciones.

Utilice la información de las siguientes secciones para controlar quién puede acceder a CloudFormation. También analizaremos cómo autorizar la creación de recursos de IAM en plantillas, conceder a las aplicaciones que se ejecutan en instancias EC2 los permisos que necesitan y utilizar credenciales de seguridad temporales para mejorar la seguridad en su entorno de AWS.

Definición de políticas de IAM basadas en identidad para CloudFormation

Para dar acceso a CloudFormation, debe crear y asignar políticas de IAM que den a sus identidades de IAM (como usuarios o roles) permiso para llamar a las acciones de API que necesitan.

Con las políticas basadas en identidad de IAM, puede especificar las acciones y recursos permitidos o denegados. CloudFormation admite acciones, claves de condiciones y recursos específicos.

Si es la primera vez que utiliza IAM, comience por familiarizarse con los elementos de una política JSON de IAM. Para obtener más información, consulte [Referencia de los elementos de la política de JSON de IAM](#) en la Guía del usuario de IAM. Para aprender a crear políticas de IAM, complete el tutorial [Crear y asociar su primera política administrada por el cliente](#) en la documentación de IAM.

Temas

- [Acciones de política para CloudFormation](#)
- [Acciones específicas de la consola de CloudFormation](#)
- [Recursos de políticas para CloudFormation](#)
- [Claves de condición de políticas para CloudFormation](#)

Acciones de política para CloudFormation

En el elemento Action de la instrucción de la política de IAM, puede especificar cualquier acción de API que CloudFormation ofrezca. Debe anteponer el nombre de la acción con la cadena `cloudformation:` en minúsculas. Por ejemplo, `cloudformation:CreateStack`, `cloudformation:CreateChangeSet` y `cloudformation:UpdateStack`.

Para especificar varias acciones en una única instrucción, sepárelas con comas del siguiente modo:

```
"Action": [ "cloudformation:action1", "cloudformation:action2" ]
```

También puede utilizar caracteres comodín para especificar varias acciones. Por ejemplo, puede especificar todas las acciones cuyos nombres comiencen con la palabra `Get` del siguiente modo:

```
"Action": "cloudformation:Get*"
```

Para ver una lista completa de las acciones asociadas al prefijo de servicio `cloudformation`, consulte [Actions, resources, and condition keys for CloudFormation](#) y [Actions, resources, and condition keys for API de control de nube de AWS](#) en la Referencia de autorización de servicios.

Ejemplos

A continuación se muestra un ejemplo de una política de permisos que concede permisos para ver las pilas de CloudFormation.

Example 1: Una política de ejemplo que concede permisos para ver la pila

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "cloudformation:DescribeStacks",
      "cloudformation:DescribeStackEvents",
      "cloudformation:DescribeStackResource",
      "cloudformation:DescribeStackResources"
    ],
    "Resource": "*"
  }]
}
```

Los usuarios que crean o eliminan pilas necesitan permisos adicionales en función de sus plantillas de pilas. Por ejemplo, si su plantilla describe una cola de Amazon SQS, los usuarios deben tener permisos para las acciones de CloudFormation y Amazon SQS, tal como se muestra en la siguiente política de ejemplo.

Example 2: Una política de ejemplo que concede acciones de creación y visualización de pila y todas las acciones Amazon SQS

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "sqs:*",
      "cloudformation:CreateStack",

```

```
        "cloudformation:DescribeStacks",
        "cloudformation:DescribeStackEvents",
        "cloudformation:DescribeStackResources",
        "cloudformation:GetTemplate",
        "cloudformation:ValidateTemplate"
    ],
    "Resource": "*"
}]
}
```

Acciones específicas de la consola de CloudFormation

Los usuarios de la consola de CloudFormation requieren permisos adicionales además de los necesarios para AWS Command Line Interface o las API de CloudFormation. Estos permisos adicionales admiten características específicas de la consola, como la carga de plantillas a buckets de Amazon S3 y listas desplegables para tipos de parámetros específicos de AWS.

Para todas las acciones enumeradas a continuación, conceda permisos a todos los recursos; no los limite a pilas o buckets específicos.

La siguiente acción se utiliza únicamente por la consola de CloudFormation y no se documenta en la referencia de la API. La acción permite a los usuarios cargar plantillas para buckets de Amazon S3.

- `cloudformation:CreateUploadBucket`

Cuando los usuarios cargan plantillas, también necesitan los siguientes permisos de Amazon S3:

- `s3:PutObject`
- `s3:ListBucket`
- `s3:GetObject`
- `s3:CreateBucket`

Para ver los valores en las listas desplegables de parámetros en las plantillas con tipos de parámetros específicos de AWS, los usuarios necesitan permisos para realizar las llamadas a la API correspondientes. Por ejemplo, se requieren los siguientes permisos cuando se utilizan estos tipos de parámetros en la plantilla:

- `ec2:DescribeKeyPairs`: necesario para el tipo de parámetro `AWS::EC2::KeyPair::KeyName`.
- `ec2:DescribeSecurityGroups`: necesario para el tipo de parámetro `AWS::EC2::SecurityGroup::Id`.
- `ec2:DescribeSubnets`: necesario para el tipo de parámetro `AWS::EC2::Subnet::Id`.
- `ec2:DescribeVpcs`: necesario para el tipo de parámetro `AWS::EC2::VPC::Id`.

Para obtener información acerca de los parámetros específicos de AWS, consulte [Definición de recursos existentes en tiempo de ejecución con tipos de parámetros proporcionados por CloudFormation](#).

Recursos de políticas para CloudFormation

En una instrucción de política de IAM, el elemento `Resource` especifica el objeto o los objetos que abarca la instrucción. Para CloudFormation, cada declaración de política de IAM se aplica a los recursos que especifique mediante su Nombre de recurso de Amazon (ARN). El formato específico de ARN depende del recurso.

Para ver una lista completa de los tipos de recursos de CloudFormation y sus ARN, consulte [Resource types defined by CloudFormation](#) en la Referencia de autorización de servicios. Para saber qué acciones puede especificar con el ARN de cada recurso, consulte [Actions defined by CloudFormation](#).

Puede especificar acciones para una pila específica, tal como se muestra en el siguiente ejemplo de política. Cuando proporcione un ARN, reemplace el *placeholder text* por la información específica del recurso.

Example 1: Una política de ejemplo que deniegue las acciones de eliminación y actualización de pila para la pila especificada.

JSON

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

```

        "Action": [
            "cloudformation:DeleteStack",
            "cloudformation:UpdateStack"
        ],
        "Resource": "arn:aws:cloudformation:us-
east-1:111122223333:stack/MyProductionStack/*"
    }
}

```

La política anterior utiliza un comodín al final del nombre de la pila a fin de denegar la pila de eliminación y la pila de actualización en el ID de pila completo (como, por ejemplo, `arn:aws:cloudformation:region:account-id:stack/MyProductionStack/abc9dbf0-43c2-11e3-a6e8-50fa526be49c`) y en el nombre de la pila (como, por ejemplo, `MyProductionStack`).

Para permitir que las transformaciones `AWS::Serverless` creen un cambio, incluya el permiso en el nivel de recursos `arn:aws:cloudformation:region:aws:transform/Serverless-2016-10-31`, tal y como se muestra en la política siguiente.

Example 2: Una política de ejemplo que permite la acción de creación de conjunto de cambios para la transformación especificada.

JSON

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "cloudformation:CreateChangeSet"
            ],
            "Resource": "arn:aws:cloudformation:us-east-1:aws:transform/
Serverless-2016-10-31"
        }
    ]
}

```

Claves de condición de políticas para CloudFormation

En una declaración de política de IAM, puede especificar opcionalmente condiciones que determinan cuándo una política está en efecto. Por ejemplo, puede definir una política que permite a los usuarios de crear una pila solo cuando se especifica una URL de plantilla determinada. Puede definir las condiciones específicas de CloudFormation y las condiciones generales de AWS como, por ejemplo, `DateLessThan`, que especifica en qué momento una política deja de estar en vigor. Para obtener más información y una lista de las condiciones generales de AWS, consulte [Condition \(Condición\)](#) en la [referencia de los elementos de las políticas de IAM](#) en la Guía del usuario de IAM.

Note

No utilice la condición general `aws:SourceIp` de AWS. CloudFormation aprovisiona recursos utilizando su propia dirección IP, no la dirección IP de la solicitud de origen. Por ejemplo, al crear una pila, CloudFormation lleva a cabo solicitudes desde su dirección IP para lanzar una instancia de Amazon EC2 o para crear un bucket de Amazon S3, en lugar de hacerlo desde la dirección IP de la llamada a `CreateStack` o el comando `create-stack`.

En la siguiente lista se describen las condiciones específicas de CloudFormation. Las condiciones se aplican únicamente cuando los usuarios crean o actualizan pilas:

`cloudformation:ChangeSetName`

Nombre de conjunto de cambios de CloudFormation que desea asociar con una política. Utilice esta condición para controlar qué conjuntos de cambios pueden ejecutar o eliminar los usuarios de .

`cloudformation:ImportResourceTypes`

Los tipos de recursos de plantilla que quiere asociar a una política como, por ejemplo, `AWS::EC2::Instance`. Utilice esta condición para controlar con qué tipos de recursos pueden trabajar los usuarios de al importar recursos a una pila. Esta condición se compara con los tipos de recursos que los usuarios declaran en el parámetro `ResourcesToImport`, que es compatible actualmente solo para solicitudes de la AWS CLI y la API. Al utilizar este parámetro, debe especificar todos los tipos de recursos que desea que los usuarios controlen durante las operaciones de importación. Para obtener más información sobre el parámetro `ResourcesToImport`, consulte la acción [CreateChangeSet](#) en la Referencia de la API de AWS CloudFormation.

Para obtener una lista de posibles `ResourcesToImport`, consulte [Tipo de recurso compatible](#).

Utilice la convención de nomenclatura de recursos de tres partes para especificar con qué tipos de recursos pueden trabajar los usuarios, desde todos los recursos de una organización hasta un tipo de recurso individual.

`organization::*`

Especifique todos los tipos de recursos para una organización determinada.

`organization::service_name::*`

Especifique todos los tipos de recursos para el servicio especificado dentro de una organización determinada.

`organization::service_name::resource_type`

Especifique un tipo de recurso específico.

Por ejemplo:

`AWS::*`

Especifique todos los tipos de recursos de AWS admitidos.

`AWS::service_name::*`

Especifique todos los recursos admitidos de un servicio de AWS específico.

`AWS::service_name::resource_type`

Especifique un tipo de recurso de AWS específico, como, por ejemplo,
`AWS::EC2::Instance` (todas las instancias de EC2).

`cloudformation:ResourceTypes`

Los tipos de recursos de plantilla como, por ejemplo *`AWS::EC2::Instance`*, que desea asociar con una política. Utilice esta condición para controlar con qué tipo de recursos pueden trabajar los usuarios de al crear o actualizar una pila. Esta condición se compara con los tipos de recursos que los usuarios declaran en el parámetro `ResourceTypes`, que es compatible actualmente solo para solicitudes de la AWS CLI y la API. Cuando se utiliza este parámetro, los usuarios deben especificar todos los tipos de recursos presentes en su plantilla. Para obtener más información sobre el parámetro `ResourceTypes`, consulte la acción [CreateStack](#) en la Referencia de la API de AWS CloudFormation.

Para obtener una lista de los tipos de recursos, consulte la [Guía de referencia de plantillas de CloudFormation](#).

Utilice la convención de nomenclatura de recursos de tres partes para especificar con qué tipos de recursos pueden trabajar los usuarios, desde todos los recursos de una organización hasta un tipo de recurso individual.

*organization::**

Especifique todos los tipos de recursos para una organización determinada.

*organization::service_name::**

Especifique todos los tipos de recursos para el servicio especificado dentro de una organización determinada.

organization::service_name::resource_type

Especifique un tipo de recurso específico.

Por ejemplo:

*AWS::**

Especifique todos los tipos de recursos de AWS admitidos.

*AWS::service_name::**

Especifique todos los recursos admitidos de un servicio de AWS específico.

AWS::service_name::resource_type

Especifique un tipo de recurso de AWS específico, como, por ejemplo,
AWS::EC2::Instance (todas las instancias de EC2).

*Alexa::ASK::**

Especifique todos los tipos de recursos en Alexa Skill Kit.

Alexa::ASK::Skill

Especifique el tipo de recursos individual de [Alexa::ASK::Skill](#).

*Custom::**

Especifique todos los recursos personalizados.

Para obtener más información, consulte [Crear una lógica de aprovisionamiento personalizada con recursos personalizados](#).

Custom::*resource_type*

Especifique un tipo de recurso personalizado específico.

Para obtener más información, consulte [Crear una lógica de aprovisionamiento personalizada con recursos personalizados](#).

cloudformation:RoleARN

El Nombre de recurso de Amazon (ARN) de un rol de servicio de IAM que desea asociar con una política. Utilice esta condición para controlar qué rol de servicio pueden utilizar los usuarios de al trabajar con pilas o conjuntos de cambios.

cloudformation:StackPolicyUrl

Una URL de política de pila de Amazon S3 que desea asociar con una política. Utilice esta condición para controlar qué políticas de pila pueden asociar los usuarios de con una pila durante una acción de creación o actualización de la pila. Para obtener más información sobre las políticas de pila, consulte [Impida actualizaciones en los recursos de la pila](#).

 Note

Para garantizar que los usuarios solo pueden crear o actualizar pilas con las políticas de pila que ha subido, establezca el bucket de S3 en modo de lectura para dichos usuarios.

cloudformation:TemplateUrl

Una URL de plantilla de Amazon S3 que desea asociar con una política. Utilice esta condición para controlar qué plantillas pueden usar los usuarios de al crear o actualizar pilas.

 Note

Para garantizar que los usuarios de solo pueden crear o actualizar pilas con las plantillas que ha subido, establezca el bucket de S3 en modo de lectura para dichos usuarios.

 Note

Las siguientes condiciones específicas para CloudFormation se aplican a los parámetros API del mismo nombre:

- `cloudformation:ChangeSetName`
- `cloudformation:RoleARN`
- `cloudformation:StackPolicyUrl`
- `cloudformation:TemplateUrl`

Por ejemplo, `cloudformation:TemplateUrl` sólo se aplica al `TemplateUrl` parámetro para `CreateStackUpdateStack`, y `CreateChangeSet` las API.

Para ver ejemplos de políticas de IAM que utilizan claves de condición para controlar el acceso, consulte [Ejemplos de políticas basadas en identidad de IAM para CloudFormation](#).

Reconocimiento de recursos de IAM en plantillas de CloudFormation

Antes de crear una pila, CloudFormation valida la plantilla. Durante la validación, CloudFormation comprueba la plantilla en busca de recursos de IAM que podría crear. Los recursos de IAM, como un usuario con acceso completo, pueden obtener acceso y modificar cualquier recurso en su cuenta de Cuenta de AWS. Por lo tanto, le sugerimos que revise los permisos asociados a cada recurso de IAM antes de continuar para no crear involuntariamente recursos con permisos escalados. Para garantizar que lo ha hecho, debe confirmar que la plantilla contiene dichos recursos, lo que concede a CloudFormation las capacidades especificadas antes de crear la pila.

Puede confirmar las capacidades de plantillas de CloudFormation a través de la consola de CloudFormation, la AWS Command Line Interface (AWS CLI) o la API:

- En la consola de CloudFormation, en la página Configurar opciones de pila de los asistentes para crear o actualizar una pila, elija Confirmando que esta plantilla pueda crear recursos de IAM.
- En la AWS CLI, cuando utilice los comandos [create-stack](#) y [update-stack](#), especifique el valor `CAPABILITY_IAM` o `CAPABILITY_NAMED_IAM` para la opción `--capabilities`. Si la plantilla incluye recursos de IAM, puede especificar cualquiera de las dos capacidades. Si la plantilla incluye nombres personalizados para recursos de IAM, debe especificar `CAPABILITY_NAMED_IAM`.
- En la API, cuando utiliza las acciones [CreateStack](#) y [UpdateStack](#) especifique `Capabilities.member.1=CAPABILITY_IAM` o `Capabilities.member.1=CAPABILITY_NAMED_IAM`. Si la plantilla incluye recursos de IAM, puede especificar cualquiera de las dos capacidades. Si la plantilla incluye nombres personalizados para recursos de IAM, debe especificar `CAPABILITY_NAMED_IAM`.

 Important

Si la plantilla contiene recursos de IAM con nombre personalizado, no cree varias pilas reutilizando la misma plantilla. Los recursos de IAM deben ser únicos globalmente en su cuenta. Si utiliza la misma plantilla para crear varias pilas en diferentes regiones, las pilas podrían compartir los mismos recursos de IAM, en lugar de que cada una tenga uno único. Los recursos compartidos entre pilas pueden tener consecuencias imprevistas de las que podría no recuperarse. Por ejemplo, si elimina o actualiza recursos de IAM compartidos en una pila, modificará involuntariamente los recursos de otras pilas.

Administración de credenciales para aplicaciones que se ejecutan en instancias de Amazon EC2

Si tiene una aplicación que se ejecuta en una instancia de Amazon EC2 y necesita llevar a cabo solicitudes a recursos de AWS, como buckets de Amazon S3 o una tabla de DynamoDB, la aplicación necesita credenciales de seguridad de AWS. Sin embargo, distribuir e incrustar credenciales de seguridad a largo plazo en cada instancia lanzada es un reto y un posible riesgo para la seguridad. En lugar de utilizar credenciales a largo plazo, como credenciales de usuario de IAM, le recomendamos que cree un rol de IAM que esté asociado a una instancia Amazon EC2 cuando se lance la instancia. Una aplicación puede obtener credenciales de seguridad temporales para la instancia Amazon EC2. No tiene que incrustar credenciales a largo plazo en la instancia. Además, para facilitar la administración de credenciales, puede especificar tan solo un rol individual para múltiples instancias Amazon EC2, no tiene que crear credenciales exclusivas para cada instancia.

Para obtener un fragmento de código de plantilla que muestre cómo lanzar una instancia con un rol, consulte [Ejemplos de plantillas de rol de IAM](#).

 Note

Las aplicaciones en las instancias que utilizan credenciales de seguridad temporales pueden llamar a cualquier acción de CloudFormation. No obstante, debido a que CloudFormation interactúa con muchos otros servicios de AWS, debe verificar que todos los servicios que desea utilizar admiten las credenciales de seguridad temporales. Para obtener una lista de los servicios que aceptan credenciales de seguridad temporales, consulte [los servicios AWS que funcionan con IAM](#) en la Guía del usuario de IAM.

Concesión de acceso temporal (acceso federado)

En algunos casos, es posible que desee conceder a los usuarios sin credenciales de AWS acceso temporal a su Cuenta de AWS. En lugar de crear y eliminar credenciales a largo plazo siempre que desee conceder acceso temporal, utilice AWS Security Token Service (AWS STS). Por ejemplo, puede utilizar roles de IAM. Desde un rol de IAM, puede crear mediante programación y, a continuación, distribuir muchas credenciales de seguridad temporales (que incluyen una clave de acceso, una clave de acceso secreta y un token de seguridad). Estas credenciales tienen una vida útil limitada, por lo que no pueden utilizarse para obtener acceso a su Cuenta de AWS cuando vencen. También puede crear varios roles de IAM con el fin de conceder a usuarios individuales diferentes niveles de permisos. Los roles de IAM resultan útiles para situaciones como identidades federadas y el inicio de sesión único.

Una identidad federada es una identidad diferenciada que puede utilizar en varios sistemas. Para los usuarios de nivel Enterprise con un sistema de identidad on-premises establecido (como, por ejemplo, LDAP o Active Directory), puede tratar todas las autenticaciones con el sistema de identidad on-premises. Cuando se ha autenticado a un usuario, proporciona credenciales de seguridad temporales del usuario o del rol de IAM apropiado. Por ejemplo, puede crear un rol de administradores y un rol de desarrolladores, en los que los administradores tengan acceso completo a la cuenta de AWS y los desarrolladores tengan permiso para trabajar únicamente con pilas de CloudFormation. Una vez que un administrador está autenticado, el administrador está autorizado para obtener credenciales de seguridad temporales del rol de administradores. Sin embargo, para los desarrolladores, pueden obtener credenciales de seguridad temporales solo del rol de desarrolladores.

También puede conceder a los usuarios federados acceso a la Consola de administración de AWS. Después de que los usuarios se autentican con su sistema de identidad on-premises, puede construir mediante programación una URL temporal que proporcione acceso directo a la Consola de administración de AWS. Cuando los usuarios utilizan la URL temporal, no es necesario que inicien sesión en AWS, dado que ya se han autenticado (inicio de sesión único). Además, dado que la URL se construye a partir de las credenciales de seguridad temporales de los usuarios, los permisos que están disponibles con esas credenciales determinan qué permisos tienen los usuarios en la Consola de administración de AWS.

Puede utilizar diferentes API de AWS STS para generar credenciales de seguridad temporales. Para obtener más información sobre qué API usar, consulte [Comparación de credenciales AWS STS](#) en la Guía del usuario de IAM.

 Important

No puede trabajar con IAM cuando utiliza credenciales de seguridad temporales que se han generado a partir de la API `GetFederationToken`. En lugar de ello, si tiene que trabajar con IAM, utilice credenciales de seguridad temporales de un rol.

CloudFormation interactúa con muchos otros servicios de AWS. Al utilizar credenciales de seguridad temporales con CloudFormation, verifique que todos los servicios que desea utilizar son compatibles con las credenciales de seguridad temporales. Para obtener una lista de los servicios que aceptan credenciales de seguridad temporales, consulte [los servicios AWS que funcionan con IAM](#) en la Guía del usuario de IAM.

Para obtener más información, consulte los siguientes recursos relacionados en la Guía del usuario de IAM:

- [Escenarios habituales en las credenciales temporales](#)
- [Permitir el acceso del agente de identidades personalizadas a la consola de AWS](#)

Ejemplos de políticas basadas en identidad de IAM para CloudFormation

De forma predeterminada, los usuarios y roles de IAM no tienen permiso para crear ni modificar los recursos de CloudFormation. Tampoco puede realizar tareas mediante la Consola de administración de AWS, la AWS Command Line Interface (AWS CLI) o la API de AWS. Un administrador de IAM puede crear políticas de IAM para conceder permisos a los usuarios para realizar acciones en los recursos que necesitan. A continuación, el administrador puede añadir las políticas de IAM a roles y los usuarios pueden asumirlos. Para obtener más información, consulte [Definición de políticas de IAM basadas en identidad para CloudFormation](#).

Los siguientes ejemplos muestran instrucciones de política que puede utilizar para permitir o denegar permisos para usar una o más acciones de CloudFormation.

Temas

- [Requerir una URL de plantilla específica](#)
- [Denegar todas las operaciones de importación de CloudFormation](#)
- [Permitir operaciones de importación para tipos de recursos específicos](#)
- [Denegar recursos de IAM en plantillas de pila](#)

- [Permitir la creación de pilas con tipos de recursos específicos](#)
- [Controlar el acceso basado en acciones de API que modifican recursos](#)
- [Restringir las operaciones de conjuntos de pilas según la región y los tipos de recursos](#)
- [Permitir todas las operaciones del generador de IaC](#)

Requerir una URL de plantilla específica

La siguiente política otorga permisos para usar únicamente la URL de la plantilla *https://s3.amazonaws.com/amzn-s3-demo-bucket/test.template* para crear o actualizar una pila.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudformation:CreateStack",
        "cloudformation:UpdateStack"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "cloudformation:TemplateUrl": [
            "https://s3.amazonaws.com/amzn-s3-demo-bucket/
test.template"
          ]
        }
      }
    }
  ]
}
```

Denegar todas las operaciones de importación de CloudFormation

La siguiente política otorga permisos para completar todas las operaciones de CloudFormation, excepto las operaciones de importación.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowAllStackOperations",
      "Effect": "Allow",
      "Action": "cloudformation:*",
      "Resource": "*"
    },
    {
      "Sid": "DenyImport",
      "Effect": "Deny",
      "Action": "cloudformation:*",
      "Resource": "*",
      "Condition": {
        "ForAnyValue:StringLike": {
          "cloudformation:ImportResourceTypes": [
            "*"
          ]
        }
      }
    }
  ]
}
```

Permitir operaciones de importación para tipos de recursos específicos

La siguiente política otorga permisos para todas las operaciones de pila, así como las operaciones de importación solo en recursos especificados (en este ejemplo, `AWS::S3::Bucket`).

JSON

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

```

    "Action": "cloudformation:*",
    "Resource": "*",
    "Condition": {
        "ForAllValues:StringEqualsIgnoreCase": {
            "cloudformation:ImportResourceTypes": [
                "AWS::S3::Bucket"
            ]
        }
    }
}
]
}

```

Denegar recursos de IAM en plantillas de pila

La siguiente política otorga permisos a los usuarios crear pilas pero deniega las solicitudes si la plantilla de la pila incluye cualquier recurso del servicio de IAM. La política también requiere que los usuarios especifiquen el parámetro `ResourceTypes`, que solo está disponible para las solicitudes de la API y la AWS CLI. Esta política utiliza declaraciones de denegación explícita de modo que si cualquier otra política otorga permisos adicionales, esta política siempre permanecerá en vigor (una declaración de denegación explícita siempre anula una declaración de permiso explícita).

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [ "cloudformation:CreateStack" ],
      "Resource": "*"
    },
    {
      "Effect": "Deny",
      "Action": [ "cloudformation:CreateStack" ],
      "Resource": "*",
      "Condition": {
        "ForAnyValue:StringLikeIfExists" : {
          "cloudformation:ResourceTypes" : [ "AWS::IAM::*" ]
        }
      }
    }
  ]
}

```

```
    },
    {
      "Effect": "Deny",
      "Action" : [ "cloudformation:CreateStack" ],
      "Resource": "*",
      "Condition": {
        "Null": {
          "cloudformation:ResourceTypes": "true"
        }
      }
    }
  ]
}
```

Permitir la creación de pilas con tipos de recursos específicos

La siguiente política es parecida al ejemplo anterior. La política otorga permisos a los usuarios crear una pila salvo si la plantilla de la pila incluye cualquier recurso del servicio de IAM. También requiere que los usuarios especifiquen el parámetro `ResourceTypes`, que solo está disponible para las solicitudes de la API y la AWS CLI. Esta política es más sencilla, pero no utiliza declaraciones de denegación explícita. Otras políticas, que conceden permisos adicionales, podrían anular esta política.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect" : "Allow",
      "Action" : [ "cloudformation:CreateStack" ],
      "Resource" : "*",
      "Condition" : {
        "ForAllValues:StringNotLikeIfExists" : {
          "cloudformation:ResourceTypes" : [ "AWS::IAM::*" ]
        },
        "Null": {
          "cloudformation:ResourceTypes": "false"
        }
      }
    }
  ]
}
```

```
]
}
```

Controlar el acceso basado en acciones de API que modifican recursos

La siguiente política otorga permisos a los usuarios filtrar el acceso por el nombre de una acción de la API de mutación de recursos. Se usa para controlar qué API pueden usar los usuarios de IAM para agregar o eliminar etiquetas en una pila o un conjunto de pilas. La operación que se utiliza para agregar o eliminar etiquetas debe agregarse como valor para la clave de condición. La siguiente política concede los permisos TagResource y UntagResource a la operación de mutación CreateStack.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "CreateActionConditionPolicyForTagUntagResources",
    "Effect": "Allow",
    "Action": [
      "cloudformation:TagResource",
      "cloudformation:UntagResource"
    ],
    "Resource": "*",
    "Condition": {
      "StringEquals": {
        "cloudformation:CreateAction": [
          "CreateStack"
        ]
      }
    }
  }]
}
```

Restringir las operaciones de conjuntos de pilas según la región y los tipos de recursos

La siguiente política otorga permisos de conjuntos de pilas administrados por servicio Un usuario con esta política solo puede llevar a cabo operaciones en conjuntos de pilas con plantillas que contengan tipos de recursos de Amazon S3 (AWS::S3::*) o el tipo de recurso

AWS::SES::ConfigurationSet. Si ha iniciado sesión en la cuenta de administración de la organización con ID 123456789012, el usuario solo puede llevar a cabo operaciones en conjuntos de pila que tengan como destino la unidad organizativa con ID *ou-1fsfsrdsdfrewr*, y solo puede llevar a cabo operaciones en el conjunto de pilas con ID *stack-set-id* que se dirija a la Cuenta de AWS con ID *987654321012*.

Las operaciones de conjunto de pila fallan si la plantilla de conjunto de pilas contiene tipos de recursos distintos de los especificados en la política, o si los destinos de implementación son unidades organizativas o ID de cuenta distintos de los especificados en la política para las cuentas de administración y los conjuntos de pilas correspondientes.

Estas restricciones de políticas solo se aplican cuando las operaciones de conjunto de pilas se dirigen a las Regiones de AWS *us-east-1*, *us-west-2* o *eu-west-2*.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudformation:*"
      ],
      "Resource": [
        "arn:aws:cloudformation:*:*:stackset/*",
        "arn:aws:cloudformation:*:*:type/resource/AWS-S3-*",
        "arn:aws:cloudformation:us-west-2:111122223333:type/resource/AWS-SES-ConfigurationSet",
        "arn:aws:cloudformation:*:111122223333:stackset-target/*/ou-1fsfsrdsdfrewr",
        "arn:aws:cloudformation:*:111122223333:stackset-target/stack-set-id/444455556666"
      ],
      "Condition": {
        "ForAllValues:StringEqualsIgnoreCase": {
          "cloudformation:TargetRegion": [
            "us-east-1",
            "us-west-2",
            "eu-west-1"
          ]
        }
      }
    }
  ]
}
```

```

    }
  }
}
]
}

```

Permitir todas las operaciones del generador de laC

La siguiente política permite el acceso a acciones de CloudFormation relacionadas con el escaneo de recursos del generador de laC y la gestión de plantillas. La primera declaración otorga permisos para describir, listar e iniciar los escaneos de recursos. También permite el acceso a los permisos adicionales necesarios (`cloudformation:GetResource`, `cloudformation:ListResources` y `cloudformation:ListTypes`) que permiten al generador de laC recuperar información sobre los recursos y los tipos de recursos disponibles. La segunda declaración otorga permisos completos para crear, eliminar, describir, listar y actualizar las plantillas generadas.

También debe conceder permisos de lectura para los servicios AWS de destino a cualquier persona que vaya a escanear recursos con el generador de laC. Para obtener más información, consulte [Permisos de IAM necesarios para escanear recursos](#).

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ResourceScanningOperations",
      "Effect": "Allow",
      "Action": [
        "cloudformation:DescribeResourceScan",
        "cloudformation:GetResource",
        "cloudformation:ListResources",
        "cloudformation:ListResourceScanRelatedResources",
        "cloudformation:ListResourceScanResources",
        "cloudformation:ListResourceScans",
        "cloudformation:ListTypes",
        "cloudformation:StartResourceScan"
      ],
      "Resource": "*"
    },
  ],
}

```

```
{
  "Sid": "TemplateGeneration",
  "Effect": "Allow",
  "Action": [
    "cloudformation:CreateGeneratedTemplate",
    "cloudformation:DeleteGeneratedTemplate",
    "cloudformation:DescribeGeneratedTemplate",
    "cloudformation:GetResource",
    "cloudformation:GetGeneratedTemplate",
    "cloudformation:ListGeneratedTemplates",
    "cloudformation:UpdateGeneratedTemplate"
  ],
  "Resource": "*"
}
```

AWS Políticas administradas de para AWS CloudFormation

Una política administrada de AWS es una política independiente que AWS crea y administra. Las políticas administradas de AWS se diseñan para ofrecer permisos para muchos casos de uso comunes, por lo que puede empezar a asignar permisos a los usuarios, grupos y roles.

Considere que es posible que las políticas administradas por AWS no concedan permisos de privilegio mínimo para los casos de uso concretos, ya que están disponibles para que las utilicen todos los clientes de AWS. Se recomienda definir [políticas administradas por el cliente](#) específicas para sus casos de uso a fin de reducir aún más los permisos.

No puede cambiar los permisos definidos en las políticas administradas AWS. Si AWS actualiza los permisos definidos en una política administrada de AWS, la actualización afecta a todas las identidades principales (usuarios, grupos y roles) a las que está adjunta la política. Lo más probable es que AWS actualice una política administrada de AWS cuando se lance un nuevo servicio AWS o las operaciones de la API nuevas estén disponibles para los servicios existentes.

Para obtener más información, consulte [Políticas administradas de AWS](#) en la Guía del usuario de IAM.

Política administrada de AWS: AWSCloudFormationFullAccess

Puede asociar AWSCloudFormationFullAccess a los usuarios, grupos y roles.

Esta política concede permisos que permiten acceso completo a todas las acciones y los recursos de CloudFormation.

Detalles de los permisos

Esta política incluye los siguientes permisos.

- `cloudformation`: permite a las entidades principales realizar todas las acciones de CloudFormation en todos los recursos.

Para ver los permisos de esta política, consulte [AWSCloudFormationFullAccess](#) en la Guía de referencia de la política administrada de AWS.

Política administrada de AWS: AWSCloudFormationReadOnlyAccess

Puede asociar `AWSCloudFormationReadOnlyAccess` a los usuarios, grupos y roles.

Esta política concede permisos que permiten acceso de solo lectura a los recursos y las acciones de CloudFormation.

Detalles de los permisos

Esta política incluye los siguientes permisos.

- `cloudformation`: permite a las entidades principales realizar acciones de solo lectura de CloudFormation, como describir pilas, enumerar recursos y ver plantillas, pero no permite crear, actualizar ni eliminar pilas.

Para ver los permisos de esta política, consulte [AWSCloudFormationReadOnlyAccess](#) en la Guía de referencia de la política administrada de AWS.

CloudFormation actualiza sus políticas administradas de AWS

Consultar los detalles sobre las actualizaciones de las políticas administradas por AWS para CloudFormation desde que este servicio comenzó a realizar el seguimiento de estos cambios. Para obtener alertas automáticas sobre cambios en esta página, suscríbase a la fuente RSS en la página de historial de documentos de CloudFormation.

Cambio	Descripción	Fecha
AWSCloudFormationReadOnlyAccess : actualización a una política existente	CloudFormation agregó permisos nuevos para permitir acciones de <code>cloudformation:BatchDescribe*</code> para operaciones de descripción por lotes.	30 de enero de 2026
AWSCloudFormationReadOnlyAccess : actualización a una política existente	CloudFormation agregó nuevos permisos para permitir acciones de <code>cloudformation:Detect*</code> para las capacidades de detección de desviaciones de pilas.	13 de noviembre de 2019
AWSCloudFormationReadOnlyAccess : actualización a una política existente	CloudFormation agregó nuevos permisos para permitir las acciones de <code>cloudformation:EstimateTemplateCost</code> , <code>cloudformation:Get*</code> y <code>cloudformation:ValidateTemplate</code> .	2 de noviembre de 2017
AWSCloudFormationFullAccess : nueva política	CloudFormation agregó una nueva política administrada de AWS que proporciona acceso total a las acciones y los recursos de CloudFormation.	26 de julio de 2019
AWSCloudFormationReadOnlyAccess : actualización a una política existente	CloudFormation agregó nuevos permisos para permitir las acciones de <code>cloudformation:DetectStackDrift</code> y <code>cloudform</code>	6 de febrero de 2019

Cambio	Descripción	Fecha
	ation:DetectStackResourceDrift para detectar la desviación de pilas.	
AWSCloudFormationReadOnlyAccess : nueva política	CloudFormation agregó una nueva política administrada de AWS que proporciona acceso de solo lectura a las acciones y los recursos de CloudFormation.	6 de febrero de 2015
CloudFormation inició el seguimiento de los cambios.	CloudFormation inició el seguimiento de los cambios de sus políticas administradas de AWS.	6 de febrero de 2015

Rol de servicio CloudFormation

Un rol de servicio es un AWS Identity and Access Management rol de (IAM) que permite a CloudFormation hacer solicitudes a recursos de una pila en su nombre. Puede especificar un rol de IAM que permita a CloudFormation crear, actualizar o eliminar los recursos de su pila. De forma predeterminada, CloudFormation utiliza una sesión temporal que se genera a partir de sus credenciales de usuario para las operaciones de pila. Si especifica un rol de servicio, CloudFormation utiliza las credenciales de ese rol.

Utilice un rol de servicio para especificar explícitamente las acciones que CloudFormation puede realizar, que es posible que no siempre sean las mismas que realiza usted u otros usuarios. Por ejemplo, puede tener privilegios administrativos, pero puede limitar el acceso a CloudFormation solo para acciones de Amazon EC2.

Puede crear el rol de servicio y su política de permisos con el servicio de IAM. Para obtener más información acerca de cómo crear un rol de servicio, consulte [Crear un rol para delegar permisos a un servicio de AWS](#) en la Guía del usuario de IAM. Especifique CloudFormation (cloudformation.amazonaws.com) como el servicio que puede asumir el rol.

Para asociar un rol de servicio a una pila, especifique el rol al crear la pila. Para obtener más información, consulte [Configurar las opciones la pila](#). También puede cambiar el rol de servicio al actualizar la pila en la consola o [DeleteStack](#) la pila a través de la API. Antes de especificar un rol de servicio, asegúrese de que tiene permiso para pasarlo (`iam:PassRole`). El permiso `iam:PassRole` especifica los roles que puede utilizar. Para obtener más información, consulte [Conceder permisos a un usuario para transferir un rol a un servicio de AWS](#) en la Guía del usuario de IAM.

Important

Cuando especifica un rol de servicio, CloudFormation siempre lo utiliza para todas las operaciones que se realizan en esa pila. No es posible eliminar un rol de servicio asociado a una pila después de crear la pila. Otros usuarios que tengan permisos para realizar operaciones en esta pila podrán usar este rol, sin importar si esos usuarios tienen o no el permiso `iam:PassRole`. Si el rol incluye permisos que el usuario no debería tener, puede escalar involuntariamente los permisos de un usuario. Asegúrese de que el rol concede privilegios mínimos. Para obtener más información, consulte [Aplicar permisos de privilegio mínimo](#) en la Guía del usuario de IAM.

Prevención de la sustitución confusa entre servicios

El problema de la sustitución confusa es un problema de seguridad en el que una entidad que no tiene permiso para realizar una acción puede obligar a una entidad con más privilegios a realizar la acción. En AWS, la suplantación entre servicios puede dar lugar al problema de la sustitución confusa. La suplantación entre servicios puede producirse cuando un servicio (el servicio que lleva a cabo las llamadas) llama a otro servicio (el servicio al que se llama). El servicio que lleva a cabo las llamadas se puede manipular para utilizar sus permisos a fin de actuar en función de los recursos de otro cliente de una manera en la que no debe tener permiso para acceder. Para evitarlo, AWS proporciona herramientas que le ayudan a proteger sus datos para todos los servicios con entidades principales de servicio a las que se les ha dado acceso a los recursos de su cuenta.

Se recomienda utilizar las claves de contexto de condición global [aws:SourceArn](#) y [aws:SourceAccount](#) en las políticas de recursos para limitar los permisos que CloudFormation le concede a otro servicio a un recurso específico, como una extensión de CloudFormation. Utiliza `aws:SourceArn` si desea que solo se asocie un recurso al acceso entre servicios. Utiliza `aws:SourceAccount` si quiere permitir que cualquier recurso de esa cuenta se asocie al uso entre servicios.

Asegúrese de que el valor de `aws:SourceArn` sea un ARN del recurso que almacena CloudFormation.

La forma más eficaz de protegerse contra el problema de la sustitución confusa es utilizar la clave de contexto de condición global de `aws:SourceArn` con el ARN completo del recurso. Si no conoce el ARN completo del recurso o si especifica varios recursos, utiliza la clave de condición de contexto global `aws:SourceArn` con comodines (*) para las partes desconocidas del ARN. Por ejemplo, `arn:aws:cloudformation*:123456789012:*`.

Si el valor de `aws:SourceArn` no contiene el ID de cuenta, debe utilizar ambas claves de contexto de condición global para limitar los permisos.

El siguiente ejemplo muestra cómo se pueden utilizar las claves contextuales de condición global `aws:SourceArn` y `aws:SourceAccount` en CloudFormation para evitar el problema del suplente confundido.

Ejemplo de política de confianza que usa las claves de condición **aws:SourceArn** y **aws:SourceAccount**

Para los servicios de registro, CloudFormation hace llamadas a AWS Security Token Service (AWS STS) para asumir un rol de servicio en su cuenta. Este rol está configurado para `ExecutionRoleArn` en la operación [RegisterType](#) y el conjunto `LogRoleArn` en la operación [LoggingConfig](#). Para obtener más información, consulte [Configure un rol de ejecución con permisos de IAM y una política de confianza para el acceso a las extensiones públicas](#).

Este ejemplo de política de confianza de rol utiliza declaraciones de condición para limitar la capacidad de `AssumeRole` en el rol de servicio a solo las acciones que afectan al grupo de extensión de CloudFormation en la cuenta especificada. Las condiciones `aws:SourceArn` y `aws:SourceAccount` se evalúan de forma independiente. Cualquier solicitud para usar el rol de servicio debe cumplir ambas condiciones.

JSON

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

```
    "Service": [
      "resources.cloudformation.amazonaws.com"
    ],
    "Action": "sts:AssumeRole",
    "Condition": {
      "StringEquals": {
        "aws:SourceAccount": "123456789012"
      },
      "ArnLike": {
        "aws:SourceArn": "arn:aws:cloudformation:us-east-1:123456789012:type/
resource/Organization-Service-Resource"
      }
    }
  }
}
```

Información adicional

Para ver ejemplos de políticas que utilizan las claves de contexto de condición global `aws:SourceArn` y `aws:SourceAccount` para un rol de servicio utilizado por StackSets, consulte [Configuración de claves globales para mitigar problemas de adjuntos confusos](#).

Para obtener más información, consulte [Actualización de una política de confianza de rol](#) en la Guía del usuario de IAM.

Solicitudes de sesiones de acceso directo (FAS) y evaluación de permisos

Al crear, actualizar y eliminar pilas de CloudFormation, los usuarios pueden especificar opcionalmente el ARN de un rol de IAM. Si no se proporciona ningún rol, CloudFormation utiliza su mecanismo de servicio predeterminado para interactuar con otros servicios de AWS. En esta situación, la persona que llama debe tener los permisos necesarios para los recursos que se administran. Como alternativa, cuando un usuario proporcione su propio rol de IAM, CloudFormation lo asumirá para interactuar con los servicios en su nombre.

Independientemente de si el usuario proporciona o no un rol de IAM, CloudFormation genera un nuevo token de FAS restringido para cada operación de recursos. Por lo tanto, las [claves de condición relacionadas con FAS](#), entre las que se incluye `aws:ViaAWSService`, se rellenan en ambas situaciones.

El uso de FAS afecta a la forma en que se evalúan las políticas de IAM durante las operaciones de CloudFormation. Al crear una pila con una plantilla que incluye los recursos que se ven afectados por las claves de condición relacionadas con FAS, es posible que se denieguen los permisos.

Política de IAM de ejemplo

Observe la política de IAM siguiente. Statement2 impedirá constantemente la creación de un recurso `AWS::KMS::Key` en CloudFormation. La restricción se aplicará de forma sistemática, independientemente de si se proporciona o no un rol de IAM durante la operación de la pila. Esto se debe a que la clave de condición `aws:ViaAWSService` siempre se establece en `true` a consecuencia del uso de FAS.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Statement1",
      "Effect": "Allow",
      "Action": [
        "kms:CreateKey"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Sid": "Statement2",
      "Effect": "Deny",
      "Action": [
        "kms:CreateKey"
      ],
      "Resource": [
        "*"
      ],
      "Condition": {
        "Bool": {
          "aws:ViaAWSService": "true"
        }
      }
    }
  ]
}
```

```
]
}
```

Plantilla de pila de ejemplo

Por ejemplo, cuando un usuario crea una pila con la siguiente plantilla de ejemplo, `aws:ViaAWSService` se establece en `true` y la política de FAS anula los permisos del rol. La creación de la pila se verá afectada por el parámetro `Statement2` de la política de IAM, que deniega la acción `CreateKey`. Esto provoca un error de permiso denegado.

```
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: '*'
```

Para obtener más información sobre FAS, consulte [Sesiones de acceso directo](#) en la Guía del usuario de IAM.

Note

La mayoría de los recursos siguen este comportamiento. Sin embargo, si la creación, actualización o eliminación de un recurso se completa correctamente o con errores de forma inesperada y su política de IAM incluye claves de condición relacionadas con FAS, es probable que el recurso en cuestión pertenezca a un pequeño subconjunto de recursos que no siguen este patrón estándar.

Registro de llamadas a la API de CloudFormation con AWS CloudTrail

CloudFormation se integra con AWS CloudTrail, un servicio que proporciona un registro de las acciones que lleva a cabo un usuario, un rol o un servicio de AWS en CloudFormation.

CloudTrail obtiene todas las llamadas a la API para CloudFormation como eventos, incluidas las llamadas procedentes de la consola de CloudFormation y de las llamadas de código a las API de CloudFormation. Si crea un registro de seguimiento, puede habilitar la entrega continua de eventos de CloudTrail a un bucket de Amazon S3, incluidos los eventos de CloudFormation. Si no configura un registro de seguimiento, puede ver los eventos más recientes de la consola de CloudTrail en el historial de eventos. Mediante la información que recopila por CloudTrail, se puede determinar la solicitud que se envió a CloudFormation, la dirección IP desde la que se realizó la solicitud, quién la realizó, cuándo se realizó y detalles adicionales.

Para obtener más información sobre CloudTrail, consulte la [Guía del usuario de AWS CloudTrail](#).

Temas

- [Información de CloudFormation en CloudTrail](#)
- [Descripción de las entradas de los archivos de registro de CloudFormation](#)

Información de CloudFormation en CloudTrail

CloudTrail se habilita en su cuenta de AWS cuando la crea. Cuando se produce una actividad en CloudFormation, esta se registra en un evento de CloudTrail junto con los eventos de los demás servicios de AWS en Event history (Historial de eventos). Se puede ver, buscar y descargar los últimos eventos de la cuenta de AWS. Para obtener más información, consulte [Ver eventos con el historial de eventos de CloudTrail](#).

Para mantener un registro continuo de eventos en la cuenta de AWS, incluidos los eventos de CloudFormation, cree un registro de seguimiento. Un registro de seguimiento permite a CloudTrail enviar archivos de registro a un bucket de Amazon S3. De manera predeterminada, cuando crea un registro de seguimiento en la consola, el registro de seguimiento se aplica a todas las regiones. El registro de seguimiento registra los eventos de todas las regiones de la partición de AWS y envía los archivos de registro al bucket de Amazon S3 especificado. También es posible configurar otros servicios de AWS para analizar en profundidad y actuar en función de los datos de eventos recopilados en los registros de CloudTrail. Para obtener más información, consulte:

- [Introducción a la creación de registros de seguimiento](#)
- [Servicios e integraciones compatibles con CloudTrail](#)
- [Configuración de notificaciones de Amazon SNS para CloudTrail](#)
- [Recibir archivos de registro de CloudTrail desde varias regiones](#) y [Recibir archivos de registro de CloudTrail desde varias cuentas](#)

CloudTrail registra todas las acciones de CloudFormation, que están documentadas en la [referencia de la API de AWS CloudFormation](#). Por ejemplo, las llamadas a las secciones `CreateStack`, `DeleteStack` y `ListStacks` generan entradas en los archivos de registro de CloudTrail.

Cada entrada de registro o evento contiene información sobre quién generó la solicitud. La información de identidad del usuario lo ayuda a determinar lo siguiente:

- Si la solicitud se realizó con las credenciales raíz o del usuario de IAM.
- Si la solicitud se realizó con credenciales de seguridad temporales de un rol o fue un usuario federado.
- Si la solicitud la realizó otro servicio de AWS.

Para más información, consulte [Elemento `userIdentity` de CloudTrail](#).

Descripción de las entradas de los archivos de registro de CloudFormation

Un registro de seguimiento es una configuración que permite la entrega de eventos como archivos de registros en un bucket de Amazon S3 que especifique. Los archivos de registro de CloudTrail pueden contener una o varias entradas de registro. Un evento representa una única solicitud de cualquier origen e incluye información sobre la operación solicitada, la fecha y la hora de la operación, los parámetros de la solicitud, etcétera. Los archivos de registro de CloudTrail no rastrean el orden en la pila de las llamadas públicas a la API, por lo que estas no aparecen en ningún orden específico.

En el ejemplo que sigue se muestra una entrada de registro de CloudTrail que ilustra la operación `CreateStack`. Una usuaria de IAM llamada Alice llevó a cabo la operación.

Note

Solo se registran los nombres de claves de los parámetros de entrada. Los valores de los parámetros no se registran.

```
{
  "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"
}
```

En el siguiente ejemplo, se muestra que Alice llamó a la operación UpdateStack en la pila my-test-stack:

```
{
  "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"
}
```

En el siguiente ejemplo, se muestra que Alice llamó a la operación ListStacks.

```
{
  "eventVersion": "1.01",
```

```
"userIdentity": {
  "type": "IAMUser",
  "principalId": "AIDAABCDEFGHijklNMOPQ",
  "arn": "arn:aws:iam::012345678910:user/Alice",
  "accountId": "012345678910",
  "accessKeyId": "AKIDEXAMPLE",
  "userName": "Alice"
},
"eventTime": "2014-03-24T21:03:16Z",
"eventSource": "cloudformation.amazonaws.com",
"eventName": "ListStacks",
"awsRegion": "us-east-1",
"sourceIPAddress": "127.0.0.1",
"userAgent": "aws-cli/1.2.11 Python/2.7.4 Linux/2.6.18-164.el5",
"requestParameters": null,
"responseElements": null,
"requestID": "b7d351d7-b397-11e3-bb75-a5b75389b02d",
"eventID": "918206d0-7281-4629-b778-b91eb0d83ce5"
}
```

En el siguiente ejemplo, se muestra que Alice llamó a la operación `DescribeStacks` en la pila `my-test-stack`.

```
{
  "eventVersion": "1.01",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDAABCDEFGHijklNMOPQ",
    "arn": "arn:aws:iam::012345678910:user/Alice",
    "accountId": "012345678910",
    "accessKeyId": "AKIDEXAMPLE",
    "userName": "Alice"
  },
  "eventTime": "2014-03-24T21:06:15Z",
  "eventSource": "cloudformation.amazonaws.com",
  "eventName": "DescribeStacks",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "127.0.0.1",
  "userAgent": "aws-cli/1.2.11 Python/2.7.4 Linux/2.6.18-164.el5",
  "requestParameters": {
    "stackName": "my-test-stack"
  },
  "responseElements": null,
}
```

```
"requestID": "224f2586-b398-11e3-bb75-a5b75389b02d",
"eventID": "9e5b2fc9-1ba8-409b-9c13-587c2ea940e2"
}
```

En el siguiente ejemplo, se muestra que Alice llamó a la operación `DeleteStack` en la pila `my-test-stack`.

```
{
  "eventVersion": "1.01",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDAABCDEFGHJKLMNOPQ",
    "arn": "arn:aws:iam::012345678910:user/Alice",
    "accountId": "012345678910",
    "accessKeyId": "AKIDEXAMPLE",
    "userName": "Alice"
  },
  "eventTime": "2014-03-24T21:07:15Z",
  "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"
}
```

Seguridad de la infraestructura en () CloudFormation

Como se trata de un servicio administrado, AWS CloudFormation está protegido por la seguridad de red global de AWS. Para obtener información sobre los servicios de seguridad de AWS y cómo AWS protege la infraestructura, consulte [Seguridad en la nube de AWS](#). Para diseñar su entorno de AWS con las prácticas recomendadas de seguridad de infraestructura, consulte [Protección de la infraestructura](#) en Portal de seguridad de AWS Well-Architected Framework.

Puede utilizar llamadas a la API publicadas en AWS para obtener acceso a CloudFormation a través de la red. Los clientes deben admitir lo siguiente:

- Seguridad de la capa de transporte (TLS). Exigimos TLS 1.2 y recomendamos TLS 1.3.
- Conjuntos de cifrado con confidencialidad directa total (PFS) como DHE (Ephemeral Diffie-Hellman) o ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). La mayoría de los sistemas modernos como Java 7 y posteriores son compatibles con estos modos.

Puede llamar a estas operaciones de la API desde cualquier ubicación de red, pero CloudFormation admite políticas de acceso basadas en recursos, que pueden incluir restricciones en función de la dirección IP de origen. También puede utilizar políticas de CloudFormation para controlar el acceso desde puntos de enlace específicos de Amazon Virtual Private Cloud (Amazon VPC) o VPC específicas. Este proceso aísla con eficacia el acceso de red a un recurso de CloudFormation determinado únicamente desde la VPC específica de la red de AWS.

Resiliencia en CloudFormation

La infraestructura global de AWS se compone de regiones de AWS y zonas de disponibilidad de AWS. Las regiones proporcionan varias zonas de disponibilidad físicamente independientes y aisladas que se encuentran conectadas mediante redes con un alto nivel de rendimiento y redundancia, además de baja latencia. Con las zonas de disponibilidad, puede diseñar y utilizar aplicaciones y bases de datos que realizan una conmutación por error automática entre las zonas sin interrupciones. Las zonas de disponibilidad tienen una mayor disponibilidad, tolerancia a errores y escalabilidad que las infraestructuras tradicionales de centros de datos únicos o múltiples.

Para obtener más información sobre las regiones y zonas de disponibilidad de AWS, consulte [Infraestructura global de AWS](#).

Validación de conformidad en CloudFormation

Para saber si un Servicio de AWS está incluido en el ámbito de programas de conformidad específicos, consulte [Servicios de AWS incluidos por programa de conformidad](#) y escoja el programa de conformidad que le interese. Para obtener información general, consulte [Programas de conformidad de AWS](#).

Puede descargar los informes de auditoría de terceros utilizando AWS Artifact. Para obtener más información, consulte [Descarga de informes en AWS Artifact](#).

Su responsabilidad de conformidad al utilizar Servicios de AWS se determina en función de la confidencialidad de los datos, los objetivos de conformidad de su empresa, así como de la legislación

y los reglamentos aplicables. Para obtener más información sobre su responsabilidad en cuanto a la conformidad al utilizar los Servicios de AWS, consulte [Documentación de seguridad de AWS](#).

Configuración y análisis de vulnerabilidades en CloudFormation

La configuración y los controles de TI son una responsabilidad compartida entre AWS y usted, nuestro cliente. Para obtener más información, consulte el AWS Modelo de responsabilidad compartida de <https://aws.amazon.com/compliance/shared-responsibility-model/>.

Prácticas recomendadas de seguridad para CloudFormation

CloudFormation proporciona un número de características de seguridad que debe tener en cuenta a la hora de desarrollar e implementar sus propias políticas de seguridad. Las siguientes prácticas recomendadas son directrices generales y no constituyen una solución de seguridad completa. Puesto que es posible que estas prácticas recomendadas no sean adecuadas o suficientes para el entorno, considérelas como consideraciones útiles en lugar de como normas.

Temas

- [Utilice IAM para controlar el acceso](#)
- [No integre credenciales en sus plantillas](#)
- [Uso de AWS CloudTrail para registrar llamadas a CloudFormation](#)

Utilice IAM para controlar el acceso

IAM es un servicio de AWS que puede utilizar para administrar los usuarios y sus permisos en AWS. Puede utilizar IAM con CloudFormation para especificar qué acciones de CloudFormation pueden realizar los usuarios, como ver plantillas de pila, crear pilas o eliminar pilas. Además, cualquiera que administre pilas de CloudFormation necesitará permisos para acceder a los recursos de dichas pilas. Por ejemplo, si los usuarios desean usar CloudFormation para lanzar, actualizar o terminar instancias Amazon EC2, deben tener permiso para llamar a las acciones de Amazon EC2 pertinentes.

En la mayoría de los casos, los usuarios requieren acceso completo para administrar todos los recursos de una plantilla. CloudFormation realiza llamadas para crear, modificar y eliminar esos recursos en su nombre. Para separar los permisos entre el usuario y el servicio de CloudFormation, utilice un rol de servicio. CloudFormation utiliza la política del rol de servicio para realizar llamadas en lugar de la política del usuario. Para obtener más información, consulte [Rol de servicio CloudFormation](#).

No integre credenciales en sus plantillas

En lugar de incrustar información confidencial en sus plantillas de CloudFormation, le recomendamos que utilice referencias dinámicas en su plantilla de pila.

Las referencias dinámicas son una forma eficaz y coherente de hacer referencia a valores externos almacenados y administrados en otros servicios, como AWS Systems Manager Parameter Store o AWS Secrets Manager. Cuando se utiliza una referencia dinámica, CloudFormation recupera el valor de la referencia especificada cuando es necesario durante las operaciones de pila y de conjunto de cambios, y pasa el valor al recurso correspondiente. Sin embargo, CloudFormation no almacena nunca el valor de referencia real. Para obtener más información, consulte [Obtención de valores almacenados en otros servicios con referencias dinámicas](#).

[AWS Secrets Manager](#) ayuda a cifrar, almacenar y recuperar de forma segura las credenciales de sus bases de datos y otros servicios. [AWS Systems Manager Parameter Store](#) proporciona un almacenamiento seguro y jerárquico para administrar los datos de configuración.

Para obtener más información sobre cómo definir parámetros de plantillas, consulte [Sintaxis de Parameters de la plantilla de CloudFormation](#).

Uso de AWS CloudTrail para registrar llamadas a CloudFormation

AWS CloudTrail realiza un seguimiento a cualquier persona que realice llamadas a la API de CloudFormation en su Cuenta de AWS. Las llamadas a la API se registran cuando se utiliza la API de CloudFormation, la consola de CloudFormation, una consola de backend o comandos de la AWS CLI de CloudFormation. Active el registro y especifique un bucket de Amazon S3 para almacenar los registros. De ese modo, si lo necesita, puede auditar quién ha hecho cada llamada a CloudFormation en su cuenta. Para obtener más información, consulte [Registro de llamadas a la API de CloudFormation con AWS CloudTrail](#).

Acceso a CloudFormation a través de un punto de conexión de la interfaz (AWS PrivateLink)

Puede usar un AWS PrivateLink para crear una conexión privada entre la VPC y CloudFormation. Puede acceder a CloudFormation como si estuviera en su VPC, sin el uso de una puerta de enlace de Internet, un dispositivo NAT, una conexión VPN o una conexión Direct Connect. Las instancias de la VPC no necesitan direcciones IP públicas para acceder a CloudFormation.

Esta conexión privada se establece mediante la creación de un punto de conexión de interfaz alimentado por AWS PrivateLink. Creamos una interfaz de red de punto de conexión en cada subred habilitada para el punto de conexión de interfaz. Se trata de interfaces de red administradas por el solicitante que sirven como punto de entrada para el tráfico destinado a CloudFormation.

CloudFormation admite la realización de llamadas a todas las acciones de la API a través del punto de conexión de interfaz.

Consideraciones para los puntos de conexión de VPC de CloudFormation

Antes de configurar un punto de conexión de interfaz, asegúrese de cumplir con los requisitos previos detallados en el tema [Acceso a un servicio de AWS a través de un punto de conexión de VPC de interfaz](#) en la Guía de AWS PrivateLink.

A continuación se aplican las siguientes consideraciones y requisitos previos adicionales al configurar un punto de conexión de interfaz para CloudFormation:

- Si tiene recursos dentro de su VPC que deben responder a una solicitud de recurso personalizado o a una condición de espera, asegúrese de que tengan acceso a los buckets necesarios de Amazon S3 específicos de CloudFormation. CloudFormation dispone de buckets de S3 en cada Región para monitorizar las respuestas a una solicitud de [recurso personalizado](#) o una [condición de espera](#). Si una plantilla incluye recursos personalizados o condiciones de espera en una VPC, la política de conexión de la VPC debe permitir a los usuarios enviar respuestas a los siguientes buckets:
 - En el caso de recursos personalizados, permita el tráfico al bucket `cloudformation-custom-resource-response-region`. Cuando se utilizan recursos personalizados, los nombres de la Región de AWS no contienen guiones. Por ejemplo, `uswest2`.
 - Para condiciones de espera, permita el tráfico al bucket `cloudformation-waitcondition-region`. Cuando se utilizan condiciones de espera, los nombres de la Región de AWS sí contienen guiones. Por ejemplo, `us-west-2`.

Si la política de puntos de enlace bloquea el tráfico a estos buckets, CloudFormation no recibirá respuestas y la operación de la pila fallará. Por ejemplo, si dispone de un recurso en una VPC en la región `us-west-2` que deben responder a una condición de espera, el recurso debe ser capaz de enviar una respuesta al bucket `cloudformation-waitcondition-us-west-2`.

Para obtener una lista de las Regiones de AWS donde CloudFormation está actualmente disponible, consulte la página de [cuotas y puntos de conexión de CloudFormation](#) en la Referencia general de Amazon Web Services.

- Los puntos de conexión de VPC actualmente no admiten las solicitudes entre regiones. Asegúrese de crear su punto de conexión en la misma región en que planea llamar a la API destinadas a CloudFormation.
- Los puntos de conexión de VPC solo admiten DNS proporcionadas por Amazon a través de Route 53. Si desea utilizar su propio DNS, puede utilizar el enrutamiento de DNS condicional. Para obtener más información, consulte [Conjuntos de opciones de DHCP en Amazon VPC](#) en la Guía del usuario de Amazon VPC.
- El grupo de seguridad asociado al punto de conexión de VPC debe permitir las conexiones entrantes en el puerto 443 desde la subred privada de la VPC.

Creación de un punto de conexión de VPC de interfaz para CloudFormation

Puede crear un punto de conexión de VPC para CloudFormation mediante la consola de Amazon VPC o la AWS Command Line Interface (AWS CLI). Para obtener más información, consulte [Create a VPC endpoint](#) (Creación de un punto de conexión de VPC) en la Guía de AWS PrivateLink.

Cree un punto de conexión para CloudFormation utilizando el siguiente nombre de servicio:

- `com.amazonaws.region.cloudformation`: crea un punto de conexión para las operaciones de la API de CloudFormation.

Si habilita DNS privado para el punto de conexión de interfaz, puede realizar solicitudes a la API para CloudFormation usando su nombre de DNS predeterminado para la región. Por ejemplo, `cloudformation.us-east-1.amazonaws.com`.

En las Regiones de AWS en donde se admiten puntos de conexión específicos de FIPS, también puede crear un punto de conexión de interfaz para CloudFormation con el siguiente nombre de servicio:

- `com.amazonaws.region.cloudformation-fips`: crea un punto de conexión para la API de CloudFormation que cumple con los [Estándares Federales de Procesamiento de la Información \(FIPS\) 140-2](#).

Para obtener una lista completa de puntos de conexión de CloudFormation, consulte los [puntos de conexión y las cuotas de CloudFormation](#) en la Referencia general de Amazon Web Services.

Creación de una política de punto de conexión de VPC para CloudFormation

Una política de puntos de conexión es un recurso de IAM que puede adjuntar al punto de conexión de su interfaz. La política de puntos de conexión predeterminada permite acceso completo a CloudFormation a través del punto de conexión de interfaz. Para controlar el acceso permitido a CloudFormation desde la VPC, adjunte una política de puntos de conexión personalizada al punto de conexión de interfaz.

Una política de punto de conexión especifica la siguiente información:

- Las entidades principales que pueden llevar a cabo acciones (Cuentas de AWS, usuarios de IAM y roles de IAM).
- Las acciones que se pueden realizar.
- El recurso en el que se pueden realizar las acciones.

Para obtener más información, consulte [Uso de políticas de punto de conexión para controlar el acceso a puntos de conexión de VPC](#) en la Guía de AWS PrivateLink.

Ejemplo: política de punto de conexión de VPC para acciones de CloudFormation

A continuación, se muestra un ejemplo de una política de puntos de conexión de CloudFormation. Cuando se asocia con un punto de conexión, esta política concede acceso a las acciones de CloudFormation mostradas para todas las entidades principales en todos los recursos. En el siguiente ejemplo, se deniega a todos los usuarios el permiso para crear pilas a través del punto de conexión de VPC y se permite el acceso completo a todas las demás acciones del servicio de CloudFormation.

```
{
  "Statement": [
    {
      "Action": "cloudformation:*",
      "Effect": "Allow",
      "Principal": "*",
      "Resource": "*"
    },
    {
      "Action": "cloudformation:CreateStack",
      "Effect": "Deny",
```

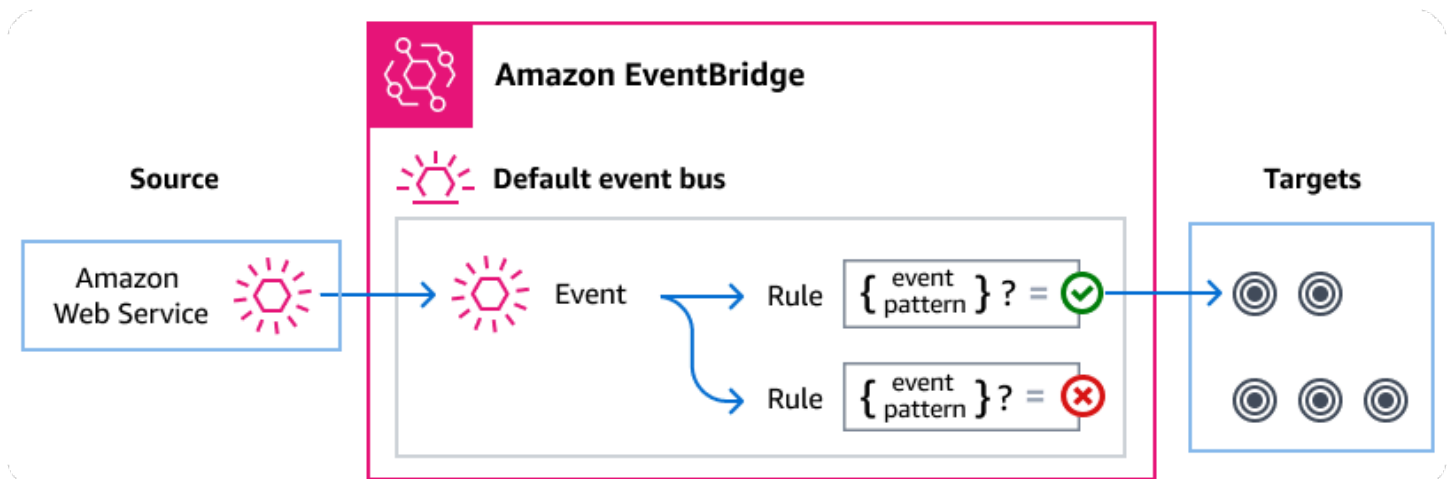
```
    "Principal": "*",  
    "Resource": "*"    
  }  
]  
}
```

Supervisión de eventos de sincronización de CloudFormation y Git con EventBridge

Amazon EventBridge es un servicio sin servidor que utiliza eventos para conectar los componentes de la aplicación entre sí, lo que facilita la creación de aplicaciones escalables basadas en eventos. La arquitectura basada en eventos es un estilo de creación de sistemas de software de acoplamiento flexible que funcionan juntos emitiendo eventos y respondiendo a ellos. Los eventos representan un cambio en un recurso o entorno.

Como ocurre con muchos servicios de AWS, CloudFormation genera y envía eventos al bus de eventos predeterminado de EventBridge, que se aprovisiona automáticamente cada Cuenta de AWS. Un bus de eventos es un enrutador que recibe eventos y los envía a cero o más destinos u objetivos. Las reglas que se especifican al bus de eventos evalúan los eventos a medida que llegan. Cada regla comprueba si un evento coincide con el patrón de evento de la regla. Si el evento coincide, el bus de eventos envía el evento a los destinos especificados.

Para obtener más información, consulte [Introducción a Amazon EventBridge](#) en la Guía del usuario de Amazon EventBridge.



Temas

- [Información general de los eventos de sincronización de CloudFormation y Git](#)
- [Permisos de Amazon EventBridge](#)
- [Creación de un patrón de eventos personalizado para una regla de EventBridge](#)
- [Referencia detallada de los eventos de CloudFormation](#)

Información general de los eventos de sincronización de CloudFormation y Git

CloudFormation envía eventos a EventBridge siempre que se lleve a cabo una operación de creación, actualización, eliminación o detección de desviaciones en una pila. CloudFormation también envía eventos a EventBridge para los cambios de estado en los conjuntos de pilas y las instancias de conjuntos de pilas. Puede utilizar las reglas de EventBridge para dirigir eventos a destinos que haya definido. Estos eventos tienen la garantía de entregarse y pueden entregarse sin ordenar.

Como los eventos de CloudFormation representan cambios en las pilas o conjuntos de pilas y sus recursos, puede utilizarlos para iniciar los flujos de trabajo asociados a los eventos respectivos. Por ejemplo:

- Cree etiquetas específicas de pilas o conjuntos de pilas en todos los recursos aprovisionados a través de CloudFormation.
- Establezca una asociación entre la pila o conjunto de pila de CloudFormation y Amazon WorkSpaces Application Manager (Amazon WAM).
- Especifique una asociación con un AppRegistry para la pila o conjunto de pila creado.

CloudFormation genera los siguientes eventos y los envía al bus de eventos predeterminado de EventBridge. Para obtener más información, consulte [???](#).

Tipo de evento	Descripción	Origen del evento
Cambio de estado del recurso	Cualquier actualización realizada en una pila que cambie las propiedades de los recursos subyacentes. Para obtener una lista completa de los tipos de recursos de AWS compatibles, consulte Referencia de tipos de recursos y propiedades de AWS.	AWS CloudFormation
Cambio de estado de la pila	Representa un cambio de estado en una pila determinada.	AWS CloudFormation

Tipo de evento	Descripción	Origen del evento
	Para obtener información detallada, consulte Códigos de estado de pilas .	
Cambio de estado de detección de desviaciones	Representa una actualización de la detección de desviaciones iniciada por el usuario en una pila determinada. Para obtener una lista completa de tipos completamente mutables e inmutables que admiten la detección de desviaciones, consulte Tipo de recurso compatible	AWS CloudFormation
Cambio de estado de StackSet	Representa un cambio de estado en un conjunto de pila determinado.	AWS CloudFormation
Cambio de estado de instancia de pila de StackSet	Representa un cambio de estado en una instancia de pila StackSet específica. Para obtener información detallada, consulte Códigos de estado de la instancia de pila .	AWS CloudFormation
Estado de la operación de StackSet	Representa un cambio de estado a una operación determinada de StackSet. Para obtener información detallada, consulte Códigos de estado de StackSets .	AWS CloudFormation

Además, la sincronización de Git de AWS CloudFormation envía eventos para los cambios de estado de las sincronizaciones de repositorios y recursos a EventBridge.

CodeConnections genera los siguientes eventos de sincronización de Git y los envía al bus de eventos predeterminado de EventBridge. Para obtener más información, consulte [???](#).

Tipo de evento	Descripción	Origen del evento
Cambio de estado de sincronización de repositorios	Representa un cambio de estado en una sincronización de repositorios de Git.	AWS CodeConnections
Cambio de estado de sincronización de recursos	Representa un cambio de estado en una sincronización de recursos de Git.	AWS CodeConnections

Permisos de Amazon EventBridge

CloudFormation no requiere permisos adicionales para entregar eventos a EventBridge. Los eventos contienen información que ya está disponible a través de las operaciones de la API de CloudFormation.

Los objetivos que especifique pueden necesitar permisos o configuraciones específicos. Para obtener más información sobre el uso de servicios específicos para los destinos, consulte [Amazon EventBridge targets](#) en la Guía del usuario de Amazon EventBridge.

Creación de un patrón de eventos personalizado para una regla de EventBridge

Puede encontrar varios patrones predefinidos en EventBridge para eventos de CloudFormation y sincronización de Git. Esto simplifica la creación de un patrón de eventos. En lugar de escribir sus propios patrones de eventos, puede seleccionar los valores de los campos de un formulario y EventBridge generará el patrón por usted. Puede crear una nueva regla mediante uno de estos patrones de eventos predefinidos o crear su propio patrón de eventos personalizado.

Cuando un servicio como CloudFormation envía un evento al bus de eventos predeterminado, EventBridge utiliza el patrón de eventos definido en la regla para determinar si el evento debe enviarse a los destinos de la regla. Un patrón de eventos coincide con los datos de los eventos de CloudFormation deseados.

Cada patrón de eventos es un objeto JSON que contiene:

- Un atributo `source` que identifica el servicio del que procede el evento. Por ejemplo, `aws.cloudformation` o `aws.codeconnections`.
- (Opcional): un atributo `detail-type` que contiene una matriz de los tipos de eventos que deben coincidir.
- (Opcional): un atributo `detail` que contiene cualquier otro dato de evento con el que coincidir.

Por ejemplo, el ID de la pila, los recursos involucrados, el estado de varios recursos y otros datos relevantes para un tipo concreto de eventos.

Por ejemplo, el siguiente patrón de eventos coincide con todos los eventos de cambio de estado de sincronización de recursos:

```
{
  "source": ["aws.cloudformation"],
  "detail-type": ["CloudFormation Resource Status Change"]
}
```

Si bien el siguiente patrón de eventos usa datos detallados de eventos para coincidir solo con los eventos de cambio de estado del recurso en los que CloudFormation crea un nuevo recurso `AWS::S3::Bucket` o `AWS::SNS::Topic`:

```
{
  "source": ["aws.cloudformation"],
  "detail-type": ["CloudFormation Resource Status Change"],
  "detail": {
    "status-details": {
      "status": ["CREATE_COMPLETE"]
    },
    "resource-type": ["AWS::S3::Bucket", "AWS::SNS::Topic"]
  }
}
```

Para obtener más información sobre la escritura de los patrones de eventos, consulte [Patrones de eventos](#) en la Guía del usuario de Amazon EventBridge.

Referencia detallada de los eventos de CloudFormation

Todos los eventos de los servicios de AWS tienen un conjunto común de campos que contienen metadatos sobre el evento, como el servicio de AWS que es el origen del evento, la hora en que

se ha generado el evento, la cuenta y la región en las que tuvo lugar el evento, etc. Para ver las definiciones de estos campos generales, consulte [Metadatos de eventos de servicios de AWS](#) en la Referencia de eventos de AWS.

Además, cada evento tiene un campo `detail` que contiene datos específicos de ese evento en particular. La siguiente referencia define los campos de detalle de los distintos eventos de CloudFormation.

Al usar EventBridge para seleccionar y administrar eventos de CloudFormation, es útil tener en cuenta lo siguiente:

- El campo `source` especifica el origen de evento.

Por ejemplo, `aws.cloudformation` o `aws.codeconnections`.

- El campo `detail-type` especifica el tipo de evento.

Por ejemplo, `CloudFormation Resource Status Change` o `CloudFormation Drift Detection Status Change`.

- El campo `detail` contiene los datos específicos de ese evento en particular.

Por ejemplo, el ID de la pila, los recursos involucrados, el estado de varios recursos y otros datos relevantes para un tipo concreto de eventos.

Para obtener más información sobre cómo construir patrones de eventos que permitan que las reglas coincidan con los eventos de CloudFormation, consulte [Patrones de eventos](#) en la Guía del usuario de Amazon EventBridge.

Para obtener más información sobre los eventos y cómo EventBridge los procesa, consulte [Eventos de EventBridge](#) en la Guía del usuario de Amazon EventBridge.

Temas

- [Detalles del evento de cambio de estado del recurso](#)
- [Detalle del evento de cambio de estado de la pila](#)
- [Detalle del evento de cambio de estado de detección de desviaciones](#)
- [Detalle del evento de cambio de estado de StackSet](#)
- [Detalle del evento de cambio de estado de la instancia de la pila de StackSet](#)
- [Detalle del evento de cambio de estado de la operación de StackSet](#)

- [Detalle del evento de cambio de estado de sincronización de repositorios](#)
- [Detalles del evento de cambio de estado de sincronización de recursos](#)

Detalles del evento de cambio de estado del recurso

A continuación, se muestran los campos de detalle de los eventos de cambio de estado del recurso.

Los campos `source` y `detail-type` se incluyen porque contienen valores específicos para los eventos.

```
{
  . . . ,
  "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

Identifica el tipo de evento.

En el caso de los eventos de estado de los recursos, este valor es `CloudFormation Resource Status Change`.

source

Identifica el servicio que generó el evento. En el caso de los eventos de CloudFormation, este valor es `aws.cloudformation`.

detail

Un objeto JSON que contiene información sobre el evento. El servicio que genera el evento determina el contenido de este campo.

En el caso de los eventos de estado de los recursos, estos datos incluyen:

`stack-id`

El ID de pila único que está asociado a la pila.

`logical-resource-id`

El nombre lógico del recurso que se especificó en la plantilla.

`physical-resource-id`

El nombre o identificador único que corresponde a un ID de instancia física de un recurso admitido por CloudFormation.

`status-details`

`status`

Estado del recurso.

`status-reason`

Motivo del estado del recurso.

`resource-type`

Tipo de recurso. Por ejemplo, `AWS::S3::Bucket`.

`client-request-token`

Un token de acceso que se usa para llamar a la API. Todos los eventos que inicia una determinada operación de pila se asignan al mismo token de solicitud de cliente, que puede utilizar para hacer un seguimiento de las operaciones. Las operaciones de la pila que se inician desde la consola utilizan el formato de token `Console-OperaciónDePila-ID`, lo que le permite identificar fácilmente la operación de la pila. Por ejemplo, si crea una pila mediante la consola, a cada evento de pila resultante se le asignará el mismo token en el siguiente formato: `Console-CreateStack-7f59c3cf-00d2-40c7-b2ff-e75db0987002`.

Example Ejemplo: evento de cambio de estado del recurso

A continuación, se muestra un ejemplo del evento de estado del recurso. Este evento detalla que CloudFormation creó correctamente el recurso solicitado, un bucket de Amazon S3, en la pila especificada.

```
{
  "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":""
  }
}
```

Detalle del evento de cambio de estado de la pila

A continuación, se muestran los campos de detalle de los eventos de cambio de estado de la pila.

Los campos `source` y `detail-type` se incluyen porque contienen valores específicos para los eventos.

```
{
  . . . ,
  "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

Identifica el tipo de evento.

En el caso los eventos de estado de la pila, este valor es `CloudFormation Stack Status Change`.

source

Identifica el servicio que generó el evento. En el caso de los eventos de CloudFormation, este valor es `aws.cloudformation`.

detail

Un objeto JSON que contiene información sobre el evento. El servicio que genera el evento determina el contenido de este campo.

En el caso de los eventos de estado de la pila, estos datos incluyen:

stack-id

El ID de pila único que está asociado a la pila.

status-details

status

Estado de la pila.

Para obtener una lista completa de los códigos de estado de la pila, consulte [Códigos de estado de pilas](#).

status-reason

Motivo del estado del recurso.

client-request-token

Un token de acceso que se usa para llamar a la API. Todos los eventos que inicia una determinada operación de pila se asignan al mismo token de solicitud de cliente, que puede utilizar para hacer un seguimiento de las operaciones. Las operaciones de la pila que se inician desde la consola utilizan el formato de token Console-OperaciónDePila-ID, lo que le permite identificar fácilmente la operación de la pila. Por ejemplo, si crea una pila mediante la consola, a cada evento de pila resultante se le asignará el mismo token en el siguiente formato: Console-CreateStack-7f59c3cf-00d2-40c7-b2ff-e75db0987002.

Example Ejemplo: evento de estado de la pila

A continuación, se muestra un ejemplo de un evento de estado de pila, en el que CloudFormation creó correctamente la pila solicitada, 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": ""
  }
}
```

Detalle del evento de cambio de estado de detección de desviaciones

A continuación, se muestran los campos de detalle de los eventos de detección de desviaciones de la pila.

Los campos `source` y `detail-type` se incluyen porque contienen valores específicos para los eventos.

```
{
  . . . ,
  "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

Identifica el tipo de evento.

En el caso de los eventos de detección de desviación de pila, este valor es `CloudFormation Drift Detection Status Change`.

source

Identifica el servicio que generó el evento. En el caso de los eventos de CloudFormation, este valor es `aws.cloudformation`.

detail

Un objeto JSON que contiene información sobre el evento. El servicio que genera el evento determina el contenido de este campo.

En el caso de los eventos de detección de desviación de pila, estos datos incluyen:

stack-id

El ID de pila único que está asociado a la pila.

`stack-drift-detection-id`

El identificador de detección de desviaciones de la pila.

`status-details`

`stack-drift-status`

Estado de desviación de la pila.

`detection-status`

Estado de la operación de detección de desviaciones.

`drift-detection-details`

`drifted-stack-resource-count`

Número de recursos desviados. Cuando el valor es -1, quiere decir que la detección de desviaciones está en progreso. Todos los demás números enteros no negativos representan el número real de recursos desviados.

`client-request-token`

Un token de acceso que se usa para llamar a la API. Todos los eventos que inicia una determinada operación de pila se asignan al mismo token de solicitud de cliente, que puede utilizar para hacer un seguimiento de las operaciones. Las operaciones de la pila que se inician desde la consola utilizan el formato de token Console-OperaciónDePila-ID, lo que le permite identificar fácilmente la operación de la pila. Por ejemplo, si crea una pila mediante la consola, a cada evento de pila resultante se le asignará el mismo token en el siguiente formato: Console-CreateStack-7f59c3cf-00d2-40c7-b2ff-e75db0987002.

Example Ejemplo: evento de detección de desviación de pila

A continuación, se muestra un ejemplo de evento de detección de desviación de pila. Este evento detalla que CloudFormation ha completado la detección de desviación de pila especificada y que la pila tiene actualmente un estado de desviación DRIFTED debido a un recurso desviado.

```
{
  "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": ""
    }
}

```

Detalle del evento de cambio de estado de StackSet

A continuación, se muestran los campos de detalle de los eventos de cambio de estado de StackSet.

Los campos `source` y `detail-type` se incluyen porque contienen valores específicos para los eventos.

```

{
    . . . ,
    "detail-type": "CloudFormation StackSet Status Change",
    "source": "aws.cloudformation",
    . . . ,
    "detail": {
        "stack-set-arn" : "string",
        "status-details": {
            "status": "string"
        }
    }
}

```

detail-type

Identifica el tipo de evento.

En el caso de los eventos de eventos de estado de StackSet, este valor es `CloudFormation StackSet Status Change`.

source

Identifica el servicio que generó el evento. En el caso de los eventos de CloudFormation, este valor es `aws.cloudformation`.

detail

Un objeto JSON que contiene información sobre el evento. El servicio que genera el evento determina el contenido de este campo.

En el caso de los eventos de evento de estado de StackSet, estos datos incluyen:

stack-set-arn

El nombre de recurso de Amazon (ARN) asociado al conjunto de la pila.

status-details

status

El estado de StackSet.

Valores válidos: ACTIVE | DELETED

Example Ejemplo: evento de cambio de estado de StackSet

A continuación, se muestra un ejemplo de un evento de cambio de estado de StackSet. Este evento detalla que CloudFormation ha eliminado el conjunto de pilas especificado.

```
{
  "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"  
    }  
  }  
}
```

Detalle del evento de cambio de estado de la instancia de la pila de StackSet

A continuación, se muestran los campos de detalle de los eventos de estado de instancia de pila de StackSet.

Los campos `source` y `detail-type` se incluyen porque contienen valores específicos para los eventos.

```
{  
  . . . ,  
  "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

Identifica el tipo de evento.

En el caso de los eventos de estado de las instancias de StackSet, este valor es `CloudFormation StackSet StackInstance Status Change`.

source

Identifica el servicio que generó el evento. En el caso de los eventos de CloudFormation, este valor es `aws.cloudformation`.

detail

Un objeto JSON que contiene información sobre el evento. El servicio que genera el evento determina el contenido de este campo.

En el caso de los eventos de estado de la instancia de StackSet, el dato incluye:

stack-set-arn

El nombre de recurso de Amazon (ARN) asociado a StackSet.

stack-id

El ID de pila único que está asociado a la instancia de pila.

action

El tipo de operación del conjunto de pilas.

Valores válidos: CREATE | UPDATE | DELETE | DETECT_DRIFT

status-details

status

El estado de la instancia de StackSet.

Para obtener más información, consulte [Códigos de estado de la instancia de pila](#).

Valores válidos: CURRENT | OUTDATED | INOPERABLE

status-reason

Motivo del estado de la instancia de StackSet.

detailed-status

El estado detallado de la instancia de StackSet detallada.

Valores válidos: CANCELLED | FAILED | FAILED_IMPORT | INOPERABLE | PENDING | RUNNING | SKIPPED_SUSPENDED_ACCOUNT | SUCCEEDED

Example Ejemplo: evento de cambio de estado de la instancia de la pila de StackSet

A continuación, se muestra un ejemplo de un evento de cambio de estado de la instancia de la pila de StackSet.

```
{
  "version": "0",
  "id": "42h6hb90-hg0w-11op-b01v-0xnhh0934z09",
  "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"
    }
  }
}
```

Detalle del evento de cambio de estado de la operación de StackSet

A continuación, se muestran los campos de detalle de los eventos de cambio de estado de la operación de StackSet.

Los campos `source` y `detail-type` se incluyen porque contienen valores específicos para los eventos.

```
{
  . . . ,
  "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

Identifica el tipo de evento.

En el caso de los eventos de estado de la operación de StackSet, este valor es `CloudFormation StackSet Operation Status Change`.

source

Identifica el servicio que generó el evento. En el caso de los eventos de CloudFormation, este valor es `aws.cloudformation`.

detail

Un objeto JSON que contiene información sobre el evento. El servicio que genera el evento determina el contenido de este campo.

En el caso de los eventos de estado de operación de StackSet, estos datos incluyen:

stack-set-arn

El nombre de recurso de Amazon (ARN) asociado a StackSet.

stack-set-operation-id

El ID único que está asociado a la operación de StackSet.

status-details

status

El estado de la operación de StackSet.

Para obtener más información, consulte [Códigos de estado de StackSets](#).

Valores válidos: `RUNNING` | `SUCCEEDED` | `FAILED` | `STOPPING` | `STOPPED` | `QUEUED`

Example Ejemplo: evento de cambio de estado de la operación de StackSet

A continuación, se muestra un ejemplo de un evento de cambio de estado de la operación de StackSet. El evento detalla que CloudFormation completó correctamente la operación solicitada en el conjunto de pilas especificado.

```
{
  "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"
    }
  }
}
```

Detalle del evento de cambio de estado de sincronización de repositorios

A continuación, se muestran los campos de detalle de los eventos de cambio de estado de sincronización de repositorios.

Los campos `source` y `detail-type` se incluyen porque contienen valores específicos para los eventos.

```
{
  . . . ,
  "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

Identifica el tipo de evento.

Para los eventos de estado de sincronización de repositorios, este valor es `Git Sync Repository Sync Status Change`.

source

Identifica el servicio que generó el evento. Para los eventos de sincronización de Git, este valor es `aws.codeconnections`.

detail

Un objeto JSON que contiene información sobre el evento. El servicio que genera el evento determina el contenido de este campo.

Para los eventos de estado de sincronización de repositorios, estos datos incluyen:

`connectionArn`

El nombre de recurso de Amazon (ARN) asociado a CodeConnections

`providerType`

El proveedor de Git conectado a CloudFormation.

Valores válidos: `GitHub` | `GitHub Enterprise` | `GitLab` | `BitBucket`

repositoryName

El nombre del repositorio de Git.

repositoryArn

El ARN asociado con el repositorio de Git

repositoryLinkId

El ID único asociado con el enlace del repositorio

ownerId

El ID único asociado con el propietario del repositorio

commit

El ID único asociado con la confirmación del repositorio.

branch

El ID único asociado con la ramificación del repositorio.

syncType

El tipo de sincronización que se está realizando

status

El estado actual de sincronización de repositorios

Valores válidos: FAILED | INITIATED | IN_PROGRESS | SUCCEEDED

previousSync

El estado de sincronización anterior al estado actual

Valores válidos: FAILED | INITIATED | IN_PROGRESS | SUCCEEDED

Example Ejemplo: Evento de cambio de estado de sincronización de repositorios

A continuación, se muestra un ejemplo de evento de cambio de estado de sincronización de repositorios. El evento detalla que CodeConnections ha sincronizado correctamente el repositorio.

```
{  
  "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",
    }
  }
}

```

Detalles del evento de cambio de estado de sincronización de recursos

A continuación, se muestran los campos de detalle de los eventos de cambio de estado de sincronización de recursos.

Los campos `source` y `detail-type` se incluyen porque contienen valores específicos para los eventos.

```

{
  . . . ,
  "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

Identifica el tipo de evento.

Para los eventos de estado de sincronización de repositorios, este valor es `Git Sync Repository Sync Status Change`.

source

Identifica el servicio que generó el evento. Para los eventos de sincronización de Git, este valor es `aws.codeconnections`.

detail

Un objeto JSON que contiene información sobre el evento. El servicio que genera el evento determina el contenido de este campo.

Para los eventos de estado de sincronización de recursos, estos datos incluyen:

providerType

El proveedor de Git conectado a CloudFormation.

Valores válidos: `GitHub` | `GitHub Enterprise` | `GitLab` | `BitBucket`

commit

El ID único asociado con la confirmación del repositorio.

repositoryName

El nombre del repositorio de Git.

branch

El ID único asociado con la ramificación del repositorio.

syncType

El tipo de sincronización que se está realizando.

syncTarget

La pila de destino para la sincronización de recursos.

status

El estado actual de sincronización de repositorios.

Valores válidos: FAILED | INITIATED | IN_PROGRESS | SUCCEEDED

previousSync

El estado de sincronización anterior al estado actual

Valores válidos: FAILED | INITIATED | IN_PROGRESS | SUCCEEDED

Example Ejemplo: Evento de cambio de estado de sincronización de recursos

A continuación, se muestra un ejemplo de evento de cambio de estado de sincronización de recursos. El evento detalla que CodeConnections ha sincronizado correctamente el recurso.

```
{
  "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"
  }
}
```

Comprensión de las cuotas de CloudFormation

Su Cuenta de AWS tiene cuotas de CloudFormation que podría necesitar conocer al crear plantillas y pilas. Al comprender estas cuotas, puede evitar los errores de limitación que exigirían que tuviera que volver a diseñar las plantillas o pilas.

En la tabla siguiente, se muestran las cuotas de CloudFormation.

Cuotas	Descripción	Valor	Estrategia de ajuste
datos de condición de espera cfn-signal	Cantidad máxima de datos que cfn-signal puede transferir.	4096 bytes	Para transferir una cantidad más grande, envíe los datos a un bucket de Amazon S3 y, a continuación, utilice cfn-signal para transferir la URL de Amazon S3 a dicho bucket.
Respuesta de recursos personalizados	Máxima cantidad de datos que puede transferir un proveedor de recursos personalizados.	4096 bytes	
Referencias dinámicas por plantilla	El número máximo de referencias dinámicas permitido en una sola plantilla de pila de CloudFormation.	60 referencias dinámicas en una plantilla de pila	
Enlaces por cuenta	Número máximo de enlaces permitidos por cuenta y región.	100 enlaces	

Cuotas	Descripción	Valor	Estrategia de ajuste
Enlaces por recurso	Número máximo de enlaces que pueden configurarse por cada recurso.	100 enlaces	
Tamaño de configuración de enlaces	Cantidad máxima de datos de configuración que un enlace puede almacenar.	204,8 KB	
Mapeos	Número máximo de asignaciones que puede declarar en su plantilla de CloudFormation.	200 asignaciones	Para especificar más mapeos, separe la plantilla en varias plantillas utilizando, por ejemplo, pilas anidadas .
Mapeo de atributos	Número máximo de atributos de asignación por cada asignación que puede declarar en su plantilla de CloudFormation.	200 atributos	Para especificar más mapeos de atributos, separe los atributos en varios mapeos.
Mapeo de nombre y mapeo de nombre de atributo	Tamaño máximo de cada nombre de asignación.	255 caracteres	
Modules	Número máximo de módulos que puede registrar en el registro de CloudFormation, por cuenta y región.	100 módulos	

Cuotas	Descripción	Valor	Estrategia de ajuste
Versiones del módulo	Número máximo de versiones que puede registrar en el registro de CloudFormation para un módulo determinado.	100 versiones	Para registrar nuevas versiones, utilice primero DerRegist erType para anular el registro de las versiones que ya no está utilizando.
Pilas anidadas	Número máximo de recursos de CloudFormation que una pila anidada puede crear, actualizar o eliminar por operación. Por ejemplo, puede tener una jerarquía de pilas anidadas con más de 2500 recursos en total, pero no puede crear, actualizar ni eliminar más de 2500 de esos recursos en una sola implementación.	2500 recursos	Divida la jerarquía de pilas en diferentes pilas.
Salidas	Número máximo de salidas que puede declarar en su plantilla de CloudFormation.	200 salidas	
Nombre de salida	Tamaño máximo de un nombre de salida.	255 caracteres	

Cuotas	Descripción	Valor	Estrategia de ajuste
Parámetros	Número máximo de parámetros que puede declarar en su plantilla de CloudFormation.	200 parámetros	Para especificar más parámetros, puede utilizar mapeos o listas para asignar múltiples valores a un único parámetro.
Nombre del parámetro	Tamaño máximo de un nombre de parámetro.	255 caracteres	
Valor del parámetro	Tamaño máximo del valor de un parámetro .	4096 bytes	Para utilizar un valor parametrizado más grande, cree varios parámetros y luego utilice la función <code>Fn::Join</code> a fin de concatenar los distintos valores en un valor único.
Recursos privados	Número máximo de recursos privados que puede registrar en el registro de CloudFormation, por cuenta y región.	50 recursos privados	
Versiones de recursos privados	Número máximo de versiones que puede registrar en el registro de CloudFormation para un recurso privado determinado.	50 recursos privados	Para registrar nuevas versiones, utilice primero DerRegist erType para anular el registro de las versiones que ya no está utilizando.

Cuotas	Descripción	Valor	Estrategia de ajuste
Recursos	Número máximo de recursos que puede declarar en su plantilla de CloudFormation.	500 recursos	Para especificar más recursos, separe la plantilla en varias plantillas utilizando, por ejemplo, pilas anidadas .
Recursos en operaciones de pila simultáneas	Número máximo de recursos que puede tener implicados en las operaciones de pila (crear, actualizar o eliminar) en una región en un momento dado.	Utilice la API DescribeAccountLimits para determinar el límite actual de una cuenta en una región específica.	
Nombre del recurso	Tamaño máximo de un nombre de recurso.	255 caracteres	
Pilas	Número máximo de pilas de CloudFormation que puede crear.	2000 pilas	Para crear más pilas, elimine las que no necesita o solicite un aumento del número máximo de pilas en su Cuenta de AWS. Para obtener más información, consulte el artículo sobre AWS Service Quotas en la Referencia general de AWS.
Nombre de pila	Tamaño máximo de un nombre de pila.	128 caracteres	

Cuotas	Descripción	Valor	Estrategia de ajuste
Conjuntos de pilas	Número máximo de conjuntos de pilas de CloudFormation que puede crear en su cuenta de administrador.	1000 conjuntos de pilas	Para crear más conjuntos de pilas, elimine los que no necesita o solicite un aumento del número máximo de conjuntos de pilas en su Cuenta de AWS. Para obtener más información, consulte el artículo sobre AWS Service Quotas en la Referencia general de AWS.
Instancias de pila	Número máximo de instancias de pila que se pueden crear por conjunto de pilas.	100 000 instancias de pilas por conjunto de pilas	Para crear más instancias de pila, elimine las que no necesita o solicite un aumento del número máximo de instancias de pila en su Cuenta de AWS. Para obtener más información, consulte el artículo sobre AWS Service Quotas en la Referencia general de AWS.

Cuotas	Descripción	Valor	Estrategia de ajuste
Operaciones de instancias de StackSets	Número máximo de instancias de pila, en todos los conjuntos de pila, en las que puede ejecutar operaciones en cada región al mismo tiempo, por cuenta de administrador.	10 000 operaciones	Este límite se aplica a todos los conjuntos de pila involucrados en una región. Incluye las instancias de pila afectadas por las operaciones de creación y actualización de conjuntos de pilas, así como la creación, actualización o eliminación de instancias de pila directamente.
Operaciones en cola de StackSets	Número máximo de operaciones en cola para un conjunto de pilas en un momento dado.	10 000 operaciones	
Pilas importadas con la operación de objeto de S3 por conjunto de pilas	Número máximo de pilas que puede importar con la operación de objeto de S3 por conjunto de pilas.	200 pilas	
Pilas importadas con la operación de indicaciones de identificaciones de pila insertadas por conjunto de pilas	Número máximo de pilas que puede importar con la operación de identificaciones de pila insertadas por conjunto de pilas.	10 pilas	

Cuotas	Descripción	Valor	Estrategia de ajuste
Tamaño del cuerpo de la plantilla en una solicitud	Tamaño máximo de un cuerpo de plantilla que puede transferir en una solicitud <code>CreateStack</code> , <code>UpdateStack</code> o <code>ValidateTemplate</code> .	51,200 bytes	Para utilizar un cuerpo de plantilla de mayor tamaño, separe la plantilla en varias plantillas utilizando, por ejemplo, pilas anidadas . O cargue la plantilla a un bucket de Amazon S3.
Tamaño del cuerpo de la plantilla en un objeto de Amazon S3	Tamaño máximo de un cuerpo de la plantilla que puede transferir en un objeto de Amazon S3 para una solicitud <code>CreateStack</code> , <code>UpdateStack</code> o <code>ValidateTemplate</code> con una URL de plantilla de Amazon S3.	1 MB	Para utilizar un cuerpo de plantilla de mayor tamaño, separe la plantilla en varias plantillas utilizando, por ejemplo, pilas anidadas . También puede utilizar la minificación para reducir el tamaño de plantilla de CloudFormation.
Descripción de la plantilla	Tamaño máximo de la descripción de la plantilla.	1024 bytes	
Versiones por enlace	Número máximo de versiones que puede crear por enlace.	100 versiones	

Disponibilidad de características

Es posible que no todas las características de CloudFormation estén disponibles en todas las regiones. Para obtener más información sobre las Regiones de AWS, consulte [Tabla de regiones de infraestructura global](#).

- Las [macros](#) no están disponibles actualmente en la siguiente región:
 - Asia-Pacífico (Yakarta)
- Actualmente, la [realización de implementaciones “azul/verde” de ECS a través de CodeDeploy con CloudFormation](#) no está disponible en las siguientes regiones:
 - África (Ciudad del Cabo)
 - Asia-Pacífico (Osaka)
 - Europa (Milán)

StackSets y macros

StackSets actualmente no es compatible con la creación o la actualización de conjuntos de pilas con permisos administrados por servicios a partir de plantillas que contienen macros. Esto incluye transformaciones, que son macros alojadas por CloudFormation. Para obtener más información sobre las macros, consulte [Macros de plantillas](#).

Solución de problemas de CloudFormation

Cuando utiliza CloudFormation, puede encontrarse con problemas al crear, actualizar o eliminar pilas de CloudFormation. Las siguientes secciones pueden ayudarle a solucionar algunos problemas habituales que puede encontrarse.

Para preguntas generales sobre CloudFormation, consulte las [Preguntas frecuentes sobre CloudFormation](#). También puede buscar respuestas y publicar preguntas en la [comunidad de CloudFormation](#) en AWS re:Post.

Temas

- [Guía para solucionar problemas](#)
- [Solución de errores](#)
- [Cómo contactar con el servicio de soporte](#)

Guía para solucionar problemas

Si CloudFormation no crea, actualiza o elimina la pila, puede ver mensajes de error o registros que le ayudarán a obtener más información sobre el problema. Las siguientes tareas describen los métodos generales para la resolución de problemas de CloudFormation. Para obtener más información sobre errores y soluciones concretos, consulte la sección [Solución de errores](#).

- Utilice los identificadores de operación y el filtrado de errores para solucionar problemas específicos. CloudFormation asigna identificadores de operación únicos a los eventos de implementación, lo que facilita el seguimiento y la correlación de los eventos relacionados. Puede filtrar los eventos de la pila por ID de operación para centrarse en fases de implementación específicas y utilizar filtros de errores para identificar rápidamente la causa principal de los errores de implementación. Consulte [Visualización de eventos de la pila por operación](#)
- Utilice la consola de CloudFormation para ver el estado de su pila. En la consola, puede ver una lista de los eventos de la pila cuando se crea, actualiza o elimina la pila. Los eventos de la pila incluyen ID de operación que le ayudan a correlacionar los eventos relacionados y a realizar un seguimiento de las operaciones individuales. Puede hacer clic en el identificador de la operación para centrarse en la solución de problemas. Desde esta operación, encuentre un evento de error y, a continuación, vea la razón del estado de dicho evento. La razón del estado podría contener un mensaje de error de CloudFormation o de un determinado servicio que puede ayudarle a

solucionar el problema. Para obtener más información acerca de la visualización de eventos de la pila con ID de operación, consulte [Visualización de eventos de la pila por operación](#).

- Para los problemas de Amazon EC2, vea los registros de `cloud-init` y `cfn`. Estos registros se publican en la instancia Amazon EC2 del directorio `/var/log/`. Estos logs capturan los procesos y salidas de comandos mientras CloudFormation configura su instancia. Para Windows, consulte el servicio EC2Configure en `%ProgramFiles%\Amazon\EC2ConfigService`, EC2 Launch en `%ProgramData%\Amazon\EC2-Windows\Launch\Logs`, EC2 Launch v2 en `%ProgramData%\Amazon\EC2Launch\log` y los registros de `cfn` en `C:\cfn\log`.

También puede configurar su plantilla de CloudFormation para que los registros se publiquen en Amazon CloudWatch, que muestra registros en la Consola de administración de AWS para que no tenga que conectarse a la instancia de Amazon EC2. Para obtener más información, consulte [Ver registros de CloudFormation en la consola](#) en el blog de administración de aplicaciones.

Solución de errores

Cuando se encuentra ante los siguientes errores con su pila de CloudFormation, puede utilizar las siguientes soluciones que le ayudarán a encontrar el origen de los problemas y corregirlos.

Temas

- [Errores al eliminar la pila](#)
- [Error de dependencia](#)
- [Conflictos AWS Config y AWS Systems Manager](#)
- [Error de análisis de parámetro al pasar una lista](#)
- [Permisos insuficientes de IAM](#)
- [Valor no válido o propiedad de recurso no admitida](#)
- [Cuota excedida](#)
- [Las pilas anidadas se han quedado atascadas en UPDATE_COMPLETE_CLEANUP_IN_PROGRESS, UPDATE_ROLLBACK_COMPLETE_CLEANUP_IN_PROGRESS o UPDATE_ROLLBACK_IN_PROGRESS](#)
- [No hay actualizaciones que realizar.](#)
- [El recurso no se estabiliza durante una operación de creación, actualización o eliminación de pila.](#)
- [El grupo de seguridad no existe en la VPC](#)

- [Error en la restauración de actualización](#)
- [La condición de espera no recibió el número necesario de señales de una instancia Amazon EC2](#)
- [Recurso eliminado de la pila pero no borrado](#)

Errores al eliminar la pila

Para resolver esta situación, pruebe lo siguiente:

- Algunos recursos deben estar vacíos para poder eliminarlos. Por ejemplo, debe eliminar todos los objetos de un bucket de Amazon S3 o eliminar todas las instancias de un grupo de seguridad de Amazon EC2 para poder eliminar el bucket o grupo de seguridad.
- Asegúrese de que tiene los permisos de IAM necesarios para eliminar los recursos de la pila. Además de los permisos de CloudFormation, debe poder utilizar los servicios subyacentes, como Amazon S3 o Amazon EC2.
- Cuando las pilas están en el estado DELETE_FAILED porque CloudFormation no ha podido eliminar un recurso, vuelva a ejecutar la [eliminación](#) con el parámetro RetainResources y especifique el recurso que CloudFormation no puede eliminar. CloudFormation elimina la pila sin eliminar el recurso conservado. Conservar los recursos resulta útil cuando no puede eliminar un recurso, como un bucket de S3 que contiene objetos que desea conservar, pero aun así desea eliminar la pila.

Después de eliminar la pila, puede eliminar manualmente los recursos conservados mediante el servicio de AWS asociado.

Como alternativa, puede considerar la posibilidad de usar la opción FORCE_DELETE_STACK con el parámetro DeletionMode. Para obtener más información sobre cómo forzar la eliminación de una pila, consulte [Eliminar una pila de la consola de CloudFormation](#).

- No puede eliminar pilas con protección de terminación habilitada. Si intenta eliminar una pila con protección de terminación habilitada, la eliminación falla y la pila, así como su estado, no cambian. Deshabilite la protección de terminación de la pila y, a continuación, realice la operación de eliminación de nuevo.

Esto incluye [pilas anidadas](#) cuyas pilas raíz tienen protección de terminación habilitada. Desactive la protección de terminación de la pila raíz y, a continuación, vuelva a llevar a cabo la operación de eliminación. Se recomienda que no elimine pilas anidadas directamente, sino como parte de la eliminación de la pila raíz y todos sus recursos.

Para obtener más información, consulte [Protección de las pilas de CloudFormation para evitar la eliminación](#).

- Para todos los demás problemas, si tiene AWS Support, puede crear un caso de Soporte. Consulte [Cómo contactar con el servicio de soporte](#).

Error de dependencia

Para resolver un error de dependencia, añada un atributo `DependsOn` a recursos que dependen de otros recursos en la plantilla. En algunos casos, debe declarar explícitamente dependencias para que CloudFormation pueda crear o eliminar los recursos en el orden correcto. Por ejemplo, si crea una IP elástica y una VPC con una puerta de enlace de Internet en la misma pila, la IP elástica debe depender de la vinculación de la puerta de enlace de Internet. Para obtener más información, consulte [DependsOn atributo](#).

Conflictos AWS Config y AWS Systems Manager

AWS Config y AWS Systems Manager pueden automatizar las tareas de administración de la infraestructura que pueden provocar conflictos con la implementación de una pila de CloudFormation. Haga lo siguiente para evitar posibles conflictos:

- Revise la configuración de AWS Config y Systems Manager en la Cuenta de AWS y la Región de AWS asociadas.
- Compruebe si hay reglas activas o documentos de automatización que puedan activarse durante una implementación de CloudFormation. Esto puede provocar conflictos o dependencias de recursos que entren en conflicto con su implementación.
- Consulte la plantilla de CloudFormation para ver si hay recursos gestionados por AWS Config y Systems Manager. Compruebe si hay posibles superposiciones o interdependencias y considere la posibilidad de ajustar la plantilla o la configuración de automatización para evitar conflictos.
- Desactive o suspenda temporalmente cualquier regla AWS Config relacionada o las automatizaciones de Systems Manager durante la implementación de CloudFormation. Recuerde restaurar las configuraciones originales después de una implementación exitosa para mantener el nivel deseado de automatización y conformidad.
- Revise los registros y los mensajes de error de CloudFormation para ver si hay referencias a problemas relacionados con AWS Config y Systems Manager y así poder identificar el origen del conflicto.

Para obtener más información sobre las reglas de AWS Config, consulte [Evaluación de recursos con reglas de Reglas de AWS Config](#).

Para obtener más información sobre las automatizaciones de Systems Manager, consulte [Automatización de AWS Systems Manager](#).

Error de análisis de parámetro al pasar una lista

Cuando utilice AWS CLI o CloudFormation para transmitir una lista, añada el carácter de escape (\) antes de cada coma. El siguiente ejemplo muestra cómo debe especificar un parámetro de entrada al utilizar la AWS CLI.

```
ParameterKey=CIDR,ParameterValue='10.10.0.0/16\,10.10.0.0/24\,10.10.1.0/24'
```

Permisos insuficientes de IAM

Al trabajar con una pila de CloudFormation no solo necesita permisos para usar CloudFormation, sino que también tiene que tener permiso para utilizar los servicios subyacentes que se describen en su plantilla. Por ejemplo, si está creando un bucket de Amazon S3 o iniciando una instancia Amazon EC2 necesita permisos para Amazon S3 o Amazon EC2. Revise la política de IAM y compruebe que dispone de los permisos necesarios antes de trabajar con pilas de CloudFormation. Para obtener más información, consulte [Control del acceso a CloudFormation con AWS Identity and Access Management](#).

Valor no válido o propiedad de recurso no admitida

Al crear o actualizar una pila de CloudFormation, la pila puede fallar debido a parámetros de entrada no válidos, nombres de propiedad de los recursos no admitidos o valores de propiedad de los recursos no admitidos. Para los parámetros de entrada, verificar que el recurso existe. Por ejemplo, cuando especifica un par de claves de Amazon EC2 o ID de la VPC, el recurso debe existir en su cuenta y en la región en la que está creando o actualizando su pila. Puede utilizar [tipos de parámetros específicos de AWS](#) para asegurarse de que utiliza valores válidos.

Para los nombres y valores de propiedades de recursos, actualice su plantilla para utilizar nombres y valores válidos. Para obtener más información, consulte la [Referencia de tipos de recursos y propiedades de AWS](#).

Cuota excedida

Verifique que no haya alcanzado una cuota de recursos. Por ejemplo, el número máximo predeterminado de instancias bajo demanda de Amazon EC2 que puede lanzar es 5. Si intenta crear un número de instancias bajo demanda de Amazon EC2 superior a la cuota de su cuenta, se produce un error en la creación de instancias y recibe el error `Status=start_failed`. Para ver las cuotas de AWS predeterminadas por servicio, consulte [AWS Service Quotas](#) en la referencia general de Referencia general de AWS.

Para conocer los límites de CloudFormation y las estrategias para realizar cambios, consulte [Comprensión de las cuotas de CloudFormation](#).

Además, durante una actualización, si se sustituye un recurso, CloudFormation crea un recurso nuevo antes de eliminar el antiguo. Este reemplazo podría poner su cuenta por encima de la cuota de recursos, lo que provocaría el error de la actualización. Puede eliminar los recursos sobrantes o solicitar un [aumento de cuota](#).

Las pilas anidadas se han quedado atascadas en **UPDATE_COMPLETE_CLEANUP_IN_PROGRESS**, **UPDATE_ROLLBACK_COMPLETE_CLEANUP_IN_PROGRESS** o **UPDATE_ROLLBACK_IN_PROGRESS**

No ha sido posible restaurar una pila anidada. Debido a posibles dependencias de recursos entre las pilas anidadas, CloudFormation no comienza la limpieza de recursos de pilas anidadas hasta que se han actualizado o se han restaurado todas las pilas anidadas. Cuando se produce un error en la restauración de una pila anidada, CloudFormation cancela todas las operaciones, independientemente del estado en que se encuentran las otras pilas anidadas. Una pila anidada que completó la actualización o la restauración pero no recibió una señal de CloudFormation para comenzar la limpieza porque otra pila anidada no se ha restaurado está en estado **UPDATE_COMPLETE_CLEANUP_IN_PROGRESS** o **UPDATE_ROLLBACK_COMPLETE_CLEANUP_IN_PROGRESS**. Una pila anidada que no se ha actualizado pero no recibió una señal para comenzar la restauración está en estado **UPDATE_ROLLBACK_IN_PROGRESS**.

Podría producirse un error en la restauración de una pila anidada debido a cambios realizados fuera de CloudFormation, cuando la plantilla de la pila no refleja de manera precisa el estado de la pila. Una pila anidada también podría fallar si un grupo de escalado automático en una pila anidada tuvo

un periodo de tiempo de espera de señal de recursos insuficiente cuando se creó o actualizó el grupo.

Para arreglar la pila, póngase en contacto con [AWS Support](#).

No hay actualizaciones que realizar.

Para actualizar una pila de CloudFormation debe enviar los cambios de los valores de los parámetros o de la plantilla a AWS CloudFormation. Sin embargo, CloudFormation no reconoce algunos cambios de la plantilla como una actualización, como cambios en una política de eliminación, una política de actualización, una declaración de condiciones o una declaración de salidas. Si necesita realizar dichos cambios sin realizar ninguno más, puede añadir o modificar un atributo de metadatos para cualquiera de sus recursos. El atributo metadata puede tener cualquier valor arbitrario, ya que CloudFormation no interpreta su contenido. Para obtener más información, consulte [Metadata atributo](#).

El recurso no se estabiliza durante una operación de creación, actualización o eliminación de pila.

Un recurso no respondió porque la operación superó el periodo de espera de CloudFormation o se interrumpió un servicio de AWS. Para las interrupciones de servicio, [compruebe](#) que el servicio de AWS pertinente se esté ejecutando y, a continuación, vuelva a intentar la operación de la pila.

Si los servicios de AWS se han estado ejecutando correctamente, compruebe si la pila contiene uno de los siguientes recursos:

- `AWS::AutoScaling::AutoScalingGroup` para operaciones de creación, actualización y eliminación.
- `AWS::CertificateManager::Certificate` para operaciones de creación
- `AWS::CloudFormation::Stack` para operaciones de creación, actualización y eliminación.
- `AWS::ElasticSearch::Domain` para operaciones de actualización.
- `AWS::RDS::DBCluster` para operaciones de creación y actualización
- `AWS::RDS::DBInstance` para operaciones de creación, actualización y eliminación.
- `AWS::Redshift::Cluster` para operaciones de actualización.

Las operaciones para estos recursos podrían tardar más que el periodo de tiempo de espera predeterminado. El periodo de tiempo de espera depende de los recursos y las credenciales que

utilice. Para ampliar el periodo de tiempo de espera, especifique un [rol de servicio](#) al realizar la operación de la pila. Si ya está usando un rol de servicio, o si la pila contiene un recurso que no aparece en la lista, contacte el [AWS Support](#).

Si la pila está en el estado `UPDATE_ROLLBACK_FAILED`, consulte [Error de restauración de actualización](#).

El grupo de seguridad no existe en la VPC

Compruebe que el grupo de seguridad existe en la VPC que especificó. Si el grupo de seguridad existe, asegúrese de que especifica el ID del grupo de seguridad y no el nombre del grupo de seguridad. Por ejemplo, el recurso `AWS::EC2::SecurityGroupIngress` tiene las propiedades `SourceSecurityGroupName` y `SourceSecurityGroupId`. Para grupos de seguridad de VPC tiene que utilizar la propiedad `SourceSecurityGroupId` y especificar el ID del grupo de seguridad.

Error en la restauración de actualización

Un recurso dependiente no puede volver a su estado original, causando un error en la restauración (estado `UPDATE_ROLLBACK_FAILED`). Por ejemplo, puede tener una pila que se está restaurando a una instancia de base de datos antigua que se eliminó fuera de CloudFormation. Dado que CloudFormation no sabe que se eliminó la base de datos, asume que la instancia de base de datos aún existe e intenta restaurarla, lo que provoca que la restauración de la actualización falle.

Dependiendo de la causa del error, puede solucionar el error manualmente y continuar con la restauración. Al continuar con la restauración, puede devolver la pila a un estado de funcionamiento (el estado `UPDATE_ROLLBACK_COMPLETE`) y, a continuación, intentar actualizar la pila de nuevo. En la siguiente lista se describen las soluciones a los errores más habituales que causan los errores de restauración de la actualización:

- Error al recibir el número necesario de señales

Utilice el comando [signal-resource](#) para enviar manualmente el número necesario de señales con éxito al recurso que las está esperando y, a continuación, continuar restaurando la actualización. Por ejemplo, durante una restauración de actualización, las instancias en un grupo de Auto Scaling podrían no señalar el éxito dentro de la duración especificada del tiempo de espera. Envíe manualmente señales de éxito al grupo de Auto Scaling. Cuando continúa con la restauración de la actualización, CloudFormation ve sus señales y procede a realizar la restauración.

- Los cambios a un recurso se hicieron fuera de CloudFormation

Sincronice manualmente recursos para que coincidan con la plantilla de pila original y, a continuación, continúe restaurando la actualización. Por ejemplo, si eliminó manualmente un recurso que CloudFormation está intentando restaurar, debe crear manualmente dicho recurso con el mismo nombre y propiedades que tenía en la pila original.

- Permisos insuficientes

Compruebe que tiene suficientes permisos de IAM para modificar recursos y, a continuación, continúe con la restauración de la actualización. Por ejemplo, su política de IAM podría permitirle crear un bucket de S3, pero no modificar el bucket. Añada las acciones de modificación a su política.

- Token de seguridad no válido

CloudFormation requiere un nuevo conjunto de credenciales. No es necesario realizar ningún cambio. Continúe restaurando la actualización, lo cual actualiza las credenciales.

- Error de limitación

Elimine los recursos que no necesita o solicite un [aumento de la cuota](#) y, a continuación, siga restaurando la actualización. Por ejemplo, si su cuota de cuenta para el número de instancias bajo demanda de EC2 es de 5 y la restauración de la actualización supera ese límite, se producirá un error.

- El recurso no se estabilizó

Un recurso no respondió porque la operación podría haber superado el periodo de espera de CloudFormation o podría haberse interrumpido un servicio de AWS. No es necesario realizar ningún cambio. Una vez que se haya completado la operación de recursos o que el servicio de AWS vuelva a estar operativo, continúe restaurando la actualización.

Para seguir restaurando una actualización, puede usar la consola de CloudFormation o la interfaz de línea de comandos de AWS (AWS CLI). Para obtener más información, consulte [Continuación de la restauración de una actualización](#).

Si ninguna de estas soluciones funciona, puede omitir los recursos que CloudFormation no puede restaurar correctamente. Para obtener más información, consulte el parámetro `ResourcesToSkip` de la operación de la API [ContinueUpdateRollback](#) en la Referencia de la API de AWS CloudFormation. CloudFormation establece el estado de los recursos especificados

como UPDATE_COMPLETE y continúa restaurando la pila. Después de que la restauración se haya completado, el estado de los recursos omitidos será incompatible con el estado de los recursos de la plantilla de pila. Antes de realizar otra actualización de pila, debe modificar los recursos o actualizar la pila o los recursos para que sean coherentes entre sí. De lo contrario, las actualizaciones de pilas posteriores podrían fallar provocando que la pila sea irrecuperable.

La condición de espera no recibió el número necesario de señales de una instancia Amazon EC2

Para resolver esta situación, pruebe lo siguiente:

- Asegúrese de que la AMI que está utilizando tiene los scripts auxiliares de CloudFormation instalados. Si la AMI no incluyen los scripts auxiliares, también puede descargarlos a la instancia. Para obtener más información, consulte [Referencia de scripts auxiliares de CloudFormation](#) en la Guía de referencia de plantillas de CloudFormation.
- Compruebe que el comando `cfn-signal` se ejecuta correctamente en la instancia. Puede ver registros, como `/var/log/cloud-init.log` o `/var/log/cfn-init.log`, para ayudarlo a depurar el lanzamiento de la instancia. Puede recuperar los registros iniciando sesión en la instancia, pero deberá [inhabilitar la restauración en caso de error](#) o de lo contrario CloudFormation eliminará la instancia después de producirse el error de creación de la pila. También puede [publicar los registros](#) en Amazon CloudWatch. Para Windows, puede ver los registros de `cfn` en `C:\cfn\log` y los registros de servicio de `EC2Configure` en `%ProgramFiles%\Amazon\EC2ConfigService`.
- Compruebe que la instancia tiene una conexión a Internet. Si la instancia está en una VPC, la instancia debería poder conectarse a Internet a través de un dispositivo NAT si se encuentra en una subred privada o a través de un puerto de enlace a Internet si no se trata de una subred pública. Para probar la conexión a Internet de la instancia, pruebe a obtener acceso a una página web pública, como por ejemplo `http://aws.amazon.com`. Por ejemplo, puede ejecutar el siguiente comando en la instancia: Debería devolver un código de estado HTTP 200.

```
curl -I https://aws.amazon.com
```

Para obtener más información acerca de cómo configurar un dispositivo NAT, consulte [NAT](#) en la Guía del usuario de Amazon VPC.

Recurso eliminado de la pila pero no borrado

Durante una actualización de pila, CloudFormation ha quitado un recurso de una pila pero no lo ha eliminado. El recurso sigue existiendo, pero ya no es accesible a través de CloudFormation. Esto puede ocurrir durante las actualizaciones de pila donde:

- CloudFormation necesita reemplazar un recurso existente, por lo que primero crea un nuevo recurso y, a continuación, intenta eliminar el antiguo recurso.
- Ha quitado el recurso de la plantilla de pila, por lo que CloudFormation intenta eliminar el recurso de la pila.

Sin embargo, puede haber casos en los que CloudFormation no puede eliminar el recurso. Por ejemplo, si el usuario no tiene permisos para eliminar un recurso de un tipo determinado.

CloudFormation intentará eliminar el antiguo recurso tres veces. Si CloudFormation no puede eliminar el recurso antiguo, lo elimina de la pila y continúa actualizando la pila. Cuando se completa la actualización de la pila, CloudFormation emite un evento de pila `UPDATE_COMPLETE`, pero incluye un `StatusReason` que indica que uno o más recursos no se pudieron eliminar. CloudFormation también emite un evento `DELETE_FAILED` para el recurso específico, con un `StatusReason` correspondiente que proporciona más detalles sobre por qué CloudFormation no pudo eliminar el recurso.

Para resolver esta situación, elimine el recurso directamente utilizando la consola o API para el servicio subyacente.

Cómo contactar con el servicio de soporte

Si tiene AWS Support, puede crear un caso de soporte técnico en <https://console.aws.amazon.com/support/home#/>. Antes de contactar con el servicio de soporte, recopile la siguiente información:

- El ID de la pila. Puede encontrar el ID de la pila en la pestaña Información general de la [consola de CloudFormation](#). Para obtener más información, consulte [Visualización de la información de la pila desde la consola de CloudFormation](#).

 Important

No realice cambios en los recursos en la pila fuera de CloudFormation. Realizar cambios en los recursos en la pila fuera de CloudFormation podría hacer que la pila entrara en un estado irrecuperable.

- Cualquier mensaje de error de la pila. Para obtener más información sobre la visualización de mensajes de error de la pila, consulte la sección [Guía para solucionar problemas](#).
- Para los problemas de Amazon EC2, reúna los registros de `cloud-init` y `cfn`. Estos registros se publican en la instancia Amazon EC2 del directorio `/var/log/`. Estos logs capturan los procesos y salidas de comandos mientras se configura su instancia. Para Windows, recopile los registros del servicio EC2Configure y de `cfn` en `%ProgramFiles%\Amazon\EC2ConfigService` y `C:\cfn\log`.

También puede buscar respuestas y publicar preguntas en la [comunidad de CloudFormation](#) en AWS re:Post.

Solución de problemas de implementaciones fallidas de pilas de CloudFormation con Amazon Q Developer

Amazon Q Developer es un asistente conversacional con tecnología de inteligencia artificial (IA) generativa que puede ayudarle a comprender, crear, extender y operar aplicaciones de AWS. Para obtener más información sobre Amazon Q Developer, consulte la [Guía del usuario de Amazon Q Developer](#).

En este documento, se describe cómo utilizar Amazon Q Developer para ayudarlo a solucionar problemas en las implementaciones fallidas de CloudFormation. Esta característica está diseñada para ayudarlo a identificar y resolver rápidamente los problemas encontrados durante la implementación de la pila.

Impulsado por Amazon Bedrock: Amazon Q está basado en Amazon Bedrock e incluye [detección automatizada de abusos](#) implementada en Amazon Bedrock para garantizar la seguridad, protección y el uso responsable de la IA.

Temas

- [Características](#)
- [Funcionamiento](#)
- [Precios](#)
- [Disponibilidad en las regiones](#)
- [Limitaciones](#)

Características

La solución de problemas de implementaciones de pilas con la solución de Amazon Q Developer proporciona las siguientes características:

- **Análisis de errores:** proporciona un análisis fácil de entender de los errores de implementación de pilas de CloudFormation.
- **Sugerencias de resolución:** ofrece instrucciones paso a paso para resolver los problemas identificados.

Funcionamiento

Cuando se produce un error al crear o actualizar una pila de CloudFormation, aparece un mensaje de error en la Consola de administración de AWS.

Puede pulsar el botón Diagnosticar con Q situado junto al mensaje de error.

Amazon Q Developer analiza el error y proporciona una explicación del problema.

Si necesita ayuda adicional, seleccione Ayuda para resolverlo para recibir los pasos detallados de resolución.

Precios

Para obtener información acerca de los precios, consulte [Precios de Amazon Q Developer](#).

Disponibilidad en las regiones

Amazon Q Developer está disponible en la mayoría de las Regiones de AWS. Para ver una lista de todas las regiones en las que Amazon Q Developer está disponible actualmente, consulte [Regiones compatibles con Amazon Q Developer](#) en la Guía del usuario de Amazon Q Developer.

Limitaciones

- Solo se puede acceder a esta característica a través de la Consola de administración de AWS.
- Esta característica cubre los 20 errores más comunes. No cubre todos los errores posibles.
- Es posible que los casos complejos de solución de problemas sigan requiriendo la intervención de Soporte.

Historial de documentos para la guía del usuario de AWS CloudFormation

En la siguiente tabla se describen cambios importantes en la Guía del usuario de AWS CloudFormation a partir de mayo de 2018. Para recibir notificaciones sobre las actualizaciones de la documentación, puede suscribirse a una fuente RSS.

Important

Las filas de la tabla que describen las actualizaciones del contenido de referencia de la plantilla a partir de mayo de 2018 se han trasladado a la nueva Referencia de plantilla de AWS CloudFormation. Para estas actualizaciones y cualquier cambio futuro, consulte el [Historial de revisión de la referencia de plantilla de AWS CloudFormation](#) en la Referencia de plantilla de AWS CloudFormation.

Cambio	Descripción	Fecha
Guía en PDF disponible	Ahora puede descargar la Guía del usuario de AWS CloudFormation como un PDF.	30 de mayo de 2025
Traslado del contenido de referencia de la plantilla a una nueva guía	CloudFormation publicó la Guía de referencia de plantillas de CloudFormation. Para obtener más información, consulte la Guía de referencia de plantillas de CloudFormation .	30 de mayo de 2025
El generador de IaC admite el escaneo parcial	Ahora puede seleccionar tipos específicos de recurso para escanear, lo que facilita la generación de plantillas de infraestructura como código	27 de marzo de 2025

(laC) a partir de los recursos existentes. Para obtener más información, consulte [Iniciar un escaneo de recursos con el generador de laC de CloudFormation](#).

[Refactorización de pilas](#)

La refactorización de pilas simplifica la reorganización de los recursos de las pilas de CloudFormation y, al mismo tiempo, conserva las propiedades y los datos de los recursos existentes. Para obtener más información, consulte [Refactorización de pilas](#).

6 de febrero de 2025

[Solución de problemas de implementaciones de pilas con Amazon Q Developer](#)

Ahora puede utilizar Amazon Q Developer para solucionar errores comunes al implementar pilas de CloudFormation. Para obtener más información, consulte [Solución de problemas de implementaciones fallidas de pilas de CloudFormation con Amazon Q Developer](#).

22 de noviembre de 2024

[Gráfico de cronograma de implementación de pilas](#)

Ahora puede ver una representación visual de la implementación de su pila. El gráfico de la línea de tiempo de la implementación de pila muestra el estado de la implementación de la pila, los estados de la implementación de los recursos individuales y las veces que han cambiado los estados de la implementación. Para obtener más información, consulte [Visualización de un cronograma de la implementación de una pila de CloudFormation](#).

11 de noviembre de 2024

[Visualización de los recursos escaneados y las plantillas generadas](#)

Ahora puede optimizar los flujos de trabajo del generador de infraestructura como código (IaC) al visualizar los detalles resumidos del escaneo y previsualizar las plantillas generadas antes de implementar su pila de infraestructura. Para obtener más información, consulte [Ver el resumen de análisis en la consola de CloudFormation](#) y [Creación de una pila de CloudFormation a partir de recursos escaneados](#).

22 de agosto de 2024

[Integración de Amazon EventBridge con la sincronización de Git de AWS CloudFormation](#)

La sincronización de Git de AWS CloudFormation publica ahora los cambios de estado de sincronización como eventos en Amazon EventBridge. Para obtener más información, consulte [Repository Sync Status Change event detail](#) y [Resource Sync Status Change event detail](#).

29 de julio de 2024

[Cómo forzar la eliminación de las pilas bloqueadas](#)

Hay disponibles dos nuevas opciones para forzar la eliminación de pilas en las operaciones de eliminación de pilas que estén bloqueadas. Ahora puede elegir entre forzar la eliminación de la pila pero conservar el recurso, o forzar la eliminación de toda la pila. Para obtener más información, consulte [Eliminar una pila de la consola de CloudFormation](#).

22 de mayo de 2024

[AWS CloudTrail Causas raíz de operaciones de pila de eventos de](#)

CloudFormation mejora la experiencia de solución de problemas de las operaciones de pila con una nueva integración de enlaces profundos de AWS CloudTrail. Esta característica vincula directamente los eventos de operaciones de pila en la consola de CloudFormation con los eventos pertinentes de CloudTrail. Para obtener más información, consulte [Determine la causa de una falla en la pila.](#)

15 de mayo de 2024

[Conjuntos de cambios a nivel de propiedad](#)

Los conjuntos de cambios a nivel de propiedad le permiten obtener una vista previa de los cambios que las implementaciones de CloudFormation realizarán en los valores de las propiedades de los recursos. Para obtener más información, consulte [Visualización de un conjunto de cambios para una pila de CloudFormation.](#)

12 de abril de 2024

[CloudFormation introduce el evento CONFIGURATION_COMPLETE](#)

Ahora puede agregar el evento CONFIGURATION_COMPLETE para permitir flujos de trabajo más rápidos que implican la creación de recursos. Para obtener más información, consulte [Comprensión de los eventos de creación de la pila de CloudFormation](#).

11 de marzo de 2024

[Generación de plantillas de AWS CloudFormation y aplicaciones de AWS CDK a partir de los recursos existentes de AWS](#)

Ahora puede generar una plantilla con los recursos aprovisionados en su cuenta que CloudFormation aún no administra. Para obtener más información, consulte [Generar plantillas de los recursos existentes con el generador IaC](#).

2 de febrero de 2024

[Modo de simultaneidad de StackSets](#)

Modo de simultaneidad es un parámetro de StackSetOperationPreferences que permite elegir cómo se comporta el nivel de simultaneidad durante las operaciones del conjunto de pilas. Para obtener más información, consulte [Elección del modo de simultaneidad para CloudFormation StackSets](#).

9 de noviembre de 2023

[Información detallada sobre la desviación de StackSets](#)

Las siguientes API le permiten ver qué instancias de pila se desviaron de la plantilla de StackSet y qué recursos se desviaron.

24 de julio de 2023

[ListStackInstanceResourceDrifts](#)

Muestra información sobre la desviación de los recursos de una instancia de pila.

[StackInstanceResourceDriftsSummary](#)

La estructura que contiene información resumida sobre las derivas de recursos para una instancia de pila.

[API de CloudFormation
StackSets para controlar el
acceso de confianza de AWS
Organizations](#)

5 de junio de 2023

CloudFormation StackSets proporciona a los clientes las siguientes API para administrar el acceso de confianza de AWS Organizations:

[ActivateOrganizationsAccess](#)

Active el acceso de confianza con AWS Organizations. Con el acceso de confianza entre StackSets y Organizations activado, la cuenta de administración tiene permisos para crear y administrar StackSets para su organización.

[DeactivateOrganizationsAccess](#)

Desactivar el acceso de confianza con AWS Organizations. Si el acceso de confianza está desactivado, la cuenta de administración no tiene permisos para crear ni administrar StackSets administrados por servicios para su organización.

[DescribeOrganizationsAccess](#)

Recupera información sobre el estado de OrganizationAccess de la cuenta. La cuenta

de administración o el administrador designado pueden llamar a esta API mediante el parámetro `CallAs`. La cuenta de administración también puede llamar a esta API sin el parámetro `CallAs`.

[DescribeStackSet API](#)

La API `DescribeStackSet` tiene un nuevo parámetro para la lista de regiones donde se implementa un conjunto de pilas determinado. Para obtener más información, consulte [DescribeStackSet](#).

1 de febrero de 2023

[Administración de eventos de StackSets con CloudFormation y Amazon EventBridge](#)

CloudFormation StackSets inicializa notificaciones de eventos a través de Amazon EventBridge. Puede activar acciones basadas en eventos después de crear, actualizar o eliminar sus conjuntos de pilas de CloudFormation. Para obtener más información, consulte [Supervisión de eventos de sincronización de CloudFormation y Git con EventBridge](#).

16 de noviembre de 2022

[Información mejorada sobre instancias de pila para operaciones de conjuntos de pilas](#)

CloudFormation StackSets proporciona información más detallada sobre instancias de pila para operaciones de conjunto de pila:

4 de noviembre de 2022

[DescribeStackSetOperation](#)

Ahora puede usar `DescribeStackSetOperation` para proporcionar el recuento de instancias de pila con errores para operaciones de conjunto de pila durante la implementación.

[ListStackInstances](#)

Ahora puede usar la opción de filtrado `LastOperationID` para listar instancias de pila para operaciones de conjunto de pila.

[Administración de eventos con CloudFormation y Amazon EventBridge](#)

Reciba notificaciones cuando se produzcan eventos específicos de CloudFormation. Para obtener más información, consulte [Supervisión de eventos de sincronización de CloudFormation y Git con EventBridge](#).

20 de julio de 2022

Cuenta

CloudFormation anuncia la disponibilidad general del tipo de filtro de cuenta, una característica que permite a los clientes limitar los destinos de implementación a cuentas individuales o incluir cuentas adicionales con las unidades organizativas proporcionadas. Para obtener más información, consulte [Objetivos de cuenta para conjuntos de pilas administrados por el servicio](#).

7 de julio de 2022

Registro de CloudFormation

CloudFormation anuncia la disponibilidad general de enlaces, una característica que permite a los clientes invocar una lógica personalizada para automatizar acciones o inspeccionar configuraciones de recursos antes de llevar a cabo operaciones de creación, actualización o eliminación de pilas. Para obtener más información, consulte la [Guía del usuario de enlaces de CloudFormation](#).

10 de febrero de 2022

[Opciones de error de pila](#)

Puede desarrollar sus aplicaciones de forma iterativa cuando se producen errores de aprovisionamiento comenzando desde el punto en que se produjo el error sin deshacer los recursos que se aprovisionaron correctamente. Al especificar las opciones de error de pila, puede solucionar los problemas de los recursos con el estado `CREATE_FAILED` o `UPDATE_FAILED`. Puede aprovisionar opciones de error para todas las implementaciones de pilas y operaciones de conjuntos de cambios. Para obtener más información, consulte [Elija cómo gestionar los errores al aprovisionar recursos](#).

30 de agosto de 2021

[Importación de pilas al conjunto de pilas](#)

Ahora puede importar pilas existentes en conjuntos de pilas nuevos o existentes. Para obtener más información, consulte [Importación de pilas en CloudFormation StackSets](#).

28 de julio de 2021

[Aumento de la cuota](#)

Ahora puede declarar un máximo predeterminado de 2000 pilas en su Cuenta de AWS. Para obtener más información, consulte [Comprensión de las cuotas de CloudFormation](#).

15 de julio de 2021

[Publicación de extensiones de terceros públicas](#)

Ahora puede utilizar las extensiones públicas de publicadores externos del mismo modo que lo haría con las extensiones de AWS. Para obtener más información, consulte [Utilización de extensiones públicas de terceros del registro de CloudFormation](#).

21 de junio de 2021

[Macros de referencia en plantillas de conjuntos de pilas](#)

Ahora StackSets es compatible con la creación o la actualización de conjuntos de pilas con permisos autoadministrados a partir de plantillas que hacen referencia a macros. Para obtener más información sobre las macros, consulte [Procesamientos personalizados en plantillas de CloudFormation con macros de plantillas](#).

14 de abril de 2021

[Uso del valor más reciente de un parámetro de Systems Manager en una referencia dinámica](#)

Ahora puede hacer que CloudFormation use la última versión de un parámetro de Systems Manager cada vez que cree o actualice una pila. Ya no es necesario indicar una versión específica. Para obtener más información, consulte [Obtención de un valor de texto sin formato del Almacén de parámetros de Systems Manager](#).

13 de abril de 2021

[Compatibilidad con módulos mediante el uso de delimitadores de puntos en nombres de recursos](#)

Ahora puede utilizar un punto como delimitador para especificar el nombre lógico completo de un recurso contenido en un módulo. Para obtener más información, consulte [Recursos del módulo de referencia en las plantillas de CloudFormation](#).

8 de abril de 2021

[CloudFormation StackSets ahora es compatible con la implementación de regiones paralelas](#)

Ahora puede optar por implementar StackSets en regiones de manera secuencial o en paralelo. Para obtener más información, consulte [Opciones de operaciones con conjuntos de pilas](#).

6 de abril de 2021

[CloudFormation StackSets ahora es compatible con el administrador delegado con AWS Organizations](#)

Además de la organización de una cuenta de administración, las cuentas del administrador delegado pueden crear y gestionar conjuntos de pilas con permisos gestionados por el servicio para su organización. Para obtener más información, consulte [Registro de una cuenta de miembro como administrador delegado](#) y [Creación de CloudFormation StackSets con permisos administrados por servicios](#).

18 de febrero de 2021

[Disponibilidad regional de CloudFormation StackSets](#)

CloudFormation StackSets ya está disponible en la región Asia-Pacífico (Osaka). Para obtener más información, consulte [Administración de pilas entre cuentas y regiones con StackSets](#).

10 de febrero de 2021

[Módulos](#)

Los módulos son una forma de empaquetar las configuraciones de recursos para su inclusión por medio de plantillas de pila, de forma transparente, administrable y repetible. Los módulos pueden encapsular configuraciones de servicios comunes y prácticas recomendadas como componentes esenciales modulares y personalizables para que pueda incluirlos en las plantillas de pila. Para obtener más información, consulte [Creación de configuraciones de recursos reutilizables que se puedan incluir en las plantillas con los módulos de CloudFormation](#).

24 de noviembre de 2020

[Conjuntos de cambios para pilas anidadas](#)

Con los conjuntos de cambios para pilas anidadas, puede previsualizar los cambios en los recursos de la aplicación y la infraestructura en toda la jerarquía de pilas anidadas y continuar con las actualizaciones cuando haya confirmado que todos los cambios son los previstos. Para obtener más información, consulte [Conjuntos de cambios para pilas anidadas](#).

18 de noviembre de 2020

Aumento de las cuotas

Se han actualizado las siguientes cuotas de AWS CloudFormation.

22 de octubre de 2020

- Ahora puede declarar un máximo de 200 asignaciones en su plantilla de AWS CloudFormation.
- Ahora puede declarar un máximo de 200 atributos de asignación para cada asignación en la plantilla de AWS CloudFormation.
- Ahora puede declarar un máximo de 200 salidas en su plantilla de AWS CloudFormation.
- Ahora puede declarar un máximo de 200 parámetros en su plantilla de AWS CloudFormation.
- Ahora puede declarar un máximo de 500 recursos en su plantilla de AWS CloudFormation.
- Ahora puede pasar un cuerpo de plantilla con un tamaño máximo de 1 MB en un objeto de Amazon S3.

[Detección de desviaciones en recursos privados](#)

CloudFormation ahora es compatible con operaciones de detección de desviaciones en una lista ampliada de recursos de AWS, así como recursos privados definidos como aprovisionables en el registro de CloudFormation. Para obtener más información, consulte [Tipos de recursos admitidos](#).

1 de octubre de 2020

[Permisos actualizados necesarios para registrar proveedores de recursos](#)

El registro de un proveedor de recursos en su cuenta ahora requiere que tenga permiso para acceder al paquete del controlador de esquema cargado en un bucket de S3 para ese proveedor de recursos. Para obtener más información, consulte [Permisos de IAM para registrar una extensión privada de terceros](#).

7 de agosto de 2020

[Compatibilidad de importación de recursos con tipos de recursos privados aprovisionables](#)

Las operaciones de importación ahora admiten tipos de recursos privados que son provisionables; es decir, cuyo tipo de aprovisionamiento es FULLY_MUTABLE o IMMUTABLE. Para obtener más información, consulte [Recursos que admiten operaciones de importación](#).

3 de junio de 2020

[Implementaciones azul/verde de ECS a través de CodeDeploy](#)

Ahora, puede usar CloudFormation para realizar implementaciones «blue/green» (azul/verde) de ECS a través de CodeDeploy. Las implementaciones azul/verde son una estrategia de implementación segura que AWS CodeDeploy proporciona para minimizar las interrupciones ocasionadas por el cambio de versiones de las aplicaciones. Para obtener más información, consulte [Realización de implementaciones azul/verde de ECS a través de CodeDeploy con CloudFormation](#).

19 de mayo de 2020

[Disponibilidad regional de CloudFormation StackSets](#)

CloudFormation StackSets ya está disponible en la región AWS GovCloud (Oeste de EE. UU.).

18 de mayo de 2020

[AWS CloudFormation
StackSets se integra con AWS
Organizations](#)

Ahora puede utilizar StackSets para administrar de forma centralizada las implementaciones en todas las cuentas de su organización o unidades organizativas (OU) específicas en AWS Organizations. También podrá habilitar las implementaciones automáticas en cualquier nueva cuenta añadida a la organización o a las unidades organizativas. StackSets administrará de forma automática los permisos necesarios para las implementaciones en las cuentas. Para obtener más información, consulte [Administración de pilas entre cuentas y regiones con StackSets](#).

11 de febrero de 2020

[Detección de desviaciones en
StackSets](#)

Ahora puede ejecutar la detección de desviaciones en un conjunto de pilas y en todas las instancias de pila que incluya. Para obtener más información, consulte [Detección de desviaciones en CloudFormation StackSets](#).

19 de noviembre de 2019

[Registro de CloudFormation ya disponible](#)

Ahora puede utilizar la consola de CloudFormation para ver los recursos privados y públicos que están disponibles para que use en su cuenta. Para obtener más información, consulte [Visualización de las extensiones disponibles y activadas en el registro de CloudFormation](#).

18 de noviembre de 2019

[Acciones de la API de registro de CloudFormation](#)

Las siguientes acciones de la API para administrar tipos en el registro de CloudFormation ya están disponibles.

18 de noviembre de 2019

[DeregisterType](#)

Elimina un tipo o una versión de tipo del uso activo en el registro de CloudFormation.

[DescribeType](#)

Devuelve información detallada sobre un tipo registrado.

[DescribeTypeRegistration](#)

Devuelve información sobre el registro de un tipo, como el estado actual y los identificadores de tipo y versión.

[ListTypeRegistrations](#)

Devuelve una lista de identificadores de solicitud de registro para el tipo especificado.

[ListTypes](#)

Devuelve información resumida sobre los tipos que se han registrado con CloudFormation.

[ListTypeVersions](#)

Devuelve información resumida sobre las versiones de un tipo.

[RegisterType](#)

Registra un tipo con el registro de CloudFormation. Si se registra un tipo, este pasa a estar disponible para usarlo en las plantillas de CloudFormation en su Cuenta de AWS.

[SetTypeDefaultVersion](#)

Especifique la versión predeterminada de un tipo. La versión predeterminada de un tipo se utilizará en las operaciones de CloudFormation.

Para obtener más información sobre el registro de CloudFormation, consulte [Administración de extensiones a través del registro de CloudFormation](#).

[Importación de recursos agregada](#)

Si ha creado un recurso de AWS fuera de la administración de CloudFormation, puede incorporar este recurso existente a la administración de CloudFormation con `resource import`. Para obtener más información, consulte [Importar recursos de AWS a una pila de CloudFormation con una importación de recursos](#).

11 de noviembre de 2019

[Aumentos de límite de conjuntos de pila](#)

Ahora puede crear un máximo de 100 conjuntos de pilas en su cuenta de administrador, crear un máximo de 2000 instancias de pila por conjunto de pilas y ejecutar un máximo de 3500 operaciones de instancias de pila en cada región al mismo tiempo, por cuenta de administrador. Para obtener más información, consulte [Comprensión de las cuotas de CloudFormation](#).

2 de agosto de 2019

[Límite para recursos en operaciones de pila simultáneas](#)

CloudFormation ahora aplica un límite de cuenta para el número de recursos en operaciones de pila simultáneas. Este límite viene determinado por la región. Para obtener más información, consulte [Comprensión de las cuotas de CloudFormation](#).

30 de abril de 2019

[Límite de operaciones de instancias de pila](#)

Para StackSets, ahora puede tener un máximo de 1500 operaciones de instancia de pila ejecutándose en una región determinada al mismo tiempo, por cuenta de administrador. Para obtener más información, consulte [Comprensión de las cuotas de CloudFormation](#).

13 de diciembre de 2018

[La capacidad CAPABILITY_AUTO_EXPAND ya está disponible](#)

Ahora puede utilizar la capacidad CAPABILITY_AUTO_EXPAND para crear o actualizar una pila directamente desde una plantilla de pilas que contiene macros, sin revisar primero los cambios resultantes en un conjunto de cambios. Para obtener más información, consulte [CreateStack](#) o [UpdateStack](#) en Referencia de la API de AWS CloudFormation.

7 de diciembre de 2018

[Se agregó la detección de desviaciones de pila](#)

Ahora puede detectar si la configuración real de la pila se ha desviado de su configuración de plantilla esperada según se define dentro de CloudFormation. Puede detectar las desviaciones en una pila completa o en recursos de la pila individuales. Para obtener más información, consulte [Detectar cambios de configuración no administrados en pilas y recursos con detección de derivación.](#)

13 de noviembre de 2018

[Ya está disponible la referencia dinámica secretsmanager](#)

Ahora puede utilizar la referencia dinámica `secretsmanager` para recuperar los valores de un secreto o la totalidad de un secreto que se almacenan en AWS Secrets Manager. Los secretos pueden ser credenciales de base de datos, contraseñas, claves de API de terceros e incluso texto arbitrario. El uso de la referencia dinámica `secretsmanager` garantiza que ni Secrets Manager ni CloudFormation registren o mantengan valores de secretos resueltos. Para obtener más información, consulte [Obtención de un secreto o un valor secreto de Secrets Manager](#).

9 de noviembre de 2018

[Macros ya disponibles](#)

Ahora puede utilizar macros para realizar un procesamiento personalizado de plantillas, desde acciones sencillas como operaciones de búsqueda y reemplazo, hasta amplias transformaciones de plantillas completas. Para obtener más información, consulte [Procesamientos personalizados en plantillas de CloudFormation con macros de plantillas](#).

6 de septiembre de 2018

[CloudFormation ahora es compatible con puntos de conexión de VPC con tecnología PrivateLink](#)

Puede utilizar un punto de conexión de VPC para crear una conexión privada entre su VPC y CloudFormation sin necesidad de obtener acceso a través de Internet, a través de una instancia NAT, una conexión de VPN o Direct Connect. Para obtener más información, consulte [Acceso a CloudFormation mediante un punto de conexión de la interfaz \(AWS PrivateLink\)](#).

22 de agosto de 2018

[Las referencias dinámicas admiten cadenas seguras](#)

Ahora puede utilizar nuevas referencias dinámicas para especificar valores que se almacenan y administran en otros servicios, incluidos parámetros de tipo SecureString de Systems Manager Parameter Store, en sus plantillas de pila. Para obtener más información, consulte [Obtención de un valor de cadena segura del Almacén de parámetros de Systems Manager](#).

16 de agosto de 2018

[Los conjuntos de pilas ya admiten funciones de ejecución personalizadas](#)

Ahora puede utilizar las funciones de ejecución personalizadas de las cuentas de destino para controlar los recursos de pila que los usuarios o grupos podrán incluir en sus conjuntos de pilas. Para obtener más información, consulte [Configuración de opciones de permisos avanzados para operaciones con conjuntos de pilas](#).

30 de mayo de 2018

[Actualizaciones selectivas de instancias de pila](#)

Ahora puede utilizar los parámetros Accounts y Regions opcionales para especificar las cuentas y regiones en las que se han de actualizar las instancias de pilas durante una operación de actualización de conjuntos de pilas. Para obtener más información, consulte [UpdateStackSet](#) en la Referencia de la API de CloudFormation.

30 de mayo de 2018

[CloudFormation ahora crea buckets de S3 con el cifrado habilitado](#)

Para los buckets de Amazon S3 que CloudFormation crea con el fin de almacenar plantillas de pila cargadas, ahora el cifrado del lado del servidor está habilitado de forma predeterminada, por lo que se cifran todos los objetos que se almacenan en esos buckets. Para más información, consulte [Creación de una pila en la consola de CloudFormation](#).

24 de mayo de 2018

[Se han agregado puntos de conexión de FIPS](#)

Ahora, CloudFormation ofrece nuevos puntos de conexión que utilizan módulos criptográficos FIPS 140-2 validados en las siguientes regiones públicas de EE. UU.: US-East-1, US-East-2, US-West-1 y US-West-2. Consulte [Puntos de conexión y cuotas de CloudFormation](#) en la Referencia general de Amazon Web Services para obtener las URL de los nuevos puntos de conexión compatibles con FIPS.

17 de mayo de 2018

Para ver las actualizaciones de la Guía del usuario de enlaces de CloudFormation, consulte el [Historial de revisión de la Guía del usuario de CloudFormation](#) en la Guía del usuario de enlaces de CloudFormation.

Actualizaciones archivadas

En la siguiente tabla, se describen los cambios importantes de cada versión de la Guía del usuario de AWS CloudFormation anteriores a mayo de 2018.

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Recursos actualizados	22 de julio de 2019	Utilice la propiedad <code>encryptionOptions</code> para especificar una Clave propiedad de AWS o una clave administrada por el cliente para agentes de Amazon MQ.	2010-05-15
Convención de nomenclatura de conjuntos de pilas	10 de abril de 2018	Las pilas de CloudFormation creadas utilizando conjuntos de pilas ahora siguen una nueva convención de nomenclatura, en la que el nombre de la pila contiene el nombre del conjunto de pilas.	2010-05-15
Recursos nuevos	10 de abril de 2018	AWS::AppSync::ApiKey Utilice el recurso <code>AWS::AppSync::ApiKey</code> para crear una clave única que puede distribuir a clientes que estén ejecutando operaciones de GraphQL con AWS AppSync. AWS::AppSync::DataSource Utilice el recurso <code>AWS::AppSync::DataSource</code> para crear orígenes de datos para solucionadores en AWS AppSync. AWS::AppSync::GraphQLApi Utilice el recurso <code>AWS::AppSync::GraphQLApi</code> para crear una nueva API de GraphQL de AWS AppSync.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
		AWS::AppSync::GraphQLSchema Utilice el recurso <code>AWS::AppSync::GraphQLSchema</code> para crear el modelo de datos para su API de GraphQL de AWS AppSync. AWS::AppSync::Resolver Utilice el recurso <code>AWS::AppSync::Resolver</code> para definir el solucionador de GraphQL lógico que asociará a campos en un esquema.	
Recurso actualizado	10 de abril de 2018	AWS::Config::ConfigurationAggregator Utilice el tipo de propiedad <code>OrganizationAggregationSource</code> para especificar las regiones de datos de AWS Config que se añadirán a un agregador de configuración de AWS Config y el rol de IAM que se debe utilizar para recuperar detalles de AWS Organizations.	2010-05-15
Recursos nuevos	4 de abril de 2018	AWS::Config::AggregationAuthorization Utilice el recurso <code>AWS::Config::AggregationAuthorization</code> para conceder permiso a una cuenta de agregador para recopilar sus datos de AWS Config. AWS::Config::ConfigurationAggregator Utilice el recurso <code>AWS::Config::ConfigurationAggregator</code> para crear un agregador de configuración para AWS Config.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Los conjuntos de pilas ahora admiten funciones de administrador personalizadas	29 de marzo de 2018	Utilice funciones de administrador personalizadas para controlar qué usuarios o grupos pueden administrar conjuntos de pilas específicos dentro de la misma cuenta de administrador. Para obtener más información, consulte Configuración de opciones de permisos avanzados para operaciones con conjuntos de pilas .	2010-05-15
Recurso nuevo	29 de marzo de 2018	AWS::EC2::LaunchTemplate Utilice el recurso <code>AWS::EC2::LaunchTemplate</code> para crear una plantilla de lanzamiento para una instancia Amazon EC2.	2010-05-15
Recursos actualizados	29 de marzo de 2018	AWS::AutoScaling::AutoScalingGroup Utilice la propiedad <code>LaunchTemplate</code> para especificar la plantilla de lanzamiento que utilizar para lanzar instancias. AWS::EC2::SpotFleet En el tipo de propiedad SpotFleetRequestConfigData , utilice la propiedad <code>LaunchTemplateConfigs</code> para describir una plantilla de lanzamiento y reemplazos.	2010-05-15
Nueva función intrínseca <code>Fn::Cidr</code>	6 de marzo de 2018	Devuelve el bloque de direcciones Cidr especificado. Para obtener más información, consulte Fn::Cidr .	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Recursos nuevos	6 de marzo de 2018	AWS::ApiGateway::VpcLink Utilice el recurso <code>AWS::ApiGateway::VpcLink</code> para especificar un enlace de VPC Gateway de API para que un <code>AWS::ApiGateway::RestApi</code> pueda obtener acceso a los recursos de una Amazon Virtual Private Cloud (VPC). AWS::GuardDuty::Master Utilice el recurso <code>AWS::GuardDuty::Master</code> para crear una cuenta principal de GuardDuty. AWS::GuardDuty::Member Utilice el recurso <code>AWS::GuardDuty::Member</code> para crear una cuenta de miembro de GuardDuty. AWS::SES::ConfigurationSet Utilice el recurso <code>AWS::SES::ConfigurationSet</code> para crear grupos de reglas que puede aplicar a los correos electrónicos que envíe. AWS::SES::ConfigurationSetEventDestination Utilice el recurso <code>AWS::SES::ConfigurationSetEventDestination</code> para especificar un destino de evento de conjunto de configuración. AWS::SES::ReceiptFilter Utilice el recurso <code>AWS::SES::ReceiptFilter</code> para especificar si se acepta o se rechaza el correo procedente de una dirección IP o un intervalo de direcciones IP.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
		<u>AWS::SES::ReceiptRule</u> Utilice el recurso <code>AWS::SES::ReceiptRule</code> para especificar qué acciones debe realizar Amazon SES cuando recibe correo en nombre de una o más direcciones de correo electrónico o dominios de su propiedad. <u>AWS::SES::ReceiptRuleSet</u> Utilice el recurso <code>AWS::SES::ReceiptRuleSet</code> para especificar un conjunto de reglas vacío para Amazon SES. <u>AWS::SES::Template</u> Utilice el recurso <code>AWS::SES::Template</code> para especificar el contenido del correo electrónico (compuesto por una línea de asunto, una parte en HTML y una parte de solo texto).	

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Recursos actualizados	6 de marzo de 2018	AWS::AutoScaling::AutoScalingGroup Utilice la propiedad <code>AutoScalingGroupName</code> para especificar el nombre del grupo de Auto Scaling. AWS::ApiGateway::RestApi Utilice la propiedad <code>ApiKeySourceType</code> para especificar el origen de la clave de API para medir las solicitudes de acuerdo con un plan de uso. Utilice la propiedad <code>MinimumCompressionSize</code> para especificar un número entero que se puede anular que se utiliza para habilitar o deshabilitar la compresión en una API. AWS::ApplicationAutoScaling::ScalingPolicy En el tipo de propiedad TargetTrackingScalingPolicyConfiguration, utilice la propiedad <code>DisableScaleIn</code> para especificar si el escalado ascendente en la política de seguimiento de destino está deshabilitado. AWS::EC2::SpotFleet En el tipo de propiedad LaunchSpecifications, utilice la propiedad <code>TagSpecifications</code> para especificar las etiquetas que aplicar durante la creación de SpotFleet. AWS::Elasticsearch::Domain Utilice el atributo <code>Arn</code> para que <code>Fn::GetAtt</code> devuelva el nombre de recurso de Amazon (ARN) del dominio.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
		El atributo <code>DomainArn</code> de <code>Fn::GetAtt</code> ha quedado obsoleto. AWS::RDS::DBCluster Utilice la propiedad <code>DBClusterIdentifier</code> para especificar el identificador del clúster de base de datos. AWS::RDS::DBCluster Utilice la propiedad <code>DBClusterIdentifier</code> para especificar el identificador del clúster de base de datos. AWS::Redshift::Cluster Utilice la propiedad <code>ClusterIdentifier</code> para especificar el identificador único del clúster. AWS::Route53::HealthCheck En el tipo de propiedad HealthCheckConfig, utilice la propiedad <code>Regions</code> para especificar las regiones desde las que desea que los comprobadores de estado de Route 53 comprueben el punto de conexión especificado. AWS::SSM::Document Utilice la propiedad <code>Tags</code> para especificar las etiquetas de recursos de CloudFormation que se van a aplicar al documento.	

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Recurso actualizado	19 de febrero de 2018	AWS::CodeBuild::Project Utilice la propiedad <code>Triggers</code> para configurar un webhook para el proyecto a fin de comenzar a recompilar automáticamente el código fuente cada vez que un cambio de código se introduce en el repositorio. Esto solo está disponible para proyectos de GitHub en CloudFormation. No está disponible para proyectos de GitHub Enterprise.	2010-05-15
Recurso actualizado	8 de febrero de 2018	AWS::DynamoDB::Table Utilice la propiedad <code>SSESpecification</code> para especificar la configuración para habilitar el cifrado del lado del servidor.	2010-05-15
Recurso actualizado	5 de febrero de 2018	AWS::CodeBuild::Project En el tipo de propiedad Source CodeBuild Project Source: • Utilice la propiedad <code>GitCloneDepth</code> para especificar la profundidad del historial que se debe descargar. • Utilice la propiedad <code>InsecureSsl</code> para hacer caso omiso de las advertencias de SSL al conectarse al repositorio del proyecto de GitHub Enterprise.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Recursos actualizados	23 de enero de 2018	AWS::AutoScaling::LifecycleHook Utilice la propiedad <code>LifecycleHookName</code> para especificar el nombre del enlace de ciclo de vida. AWS::DynamoDB::Table La propiedad <code>AttributeDefinitions</code> ahora requiere la sustitución al actualizarla. AWS::EC2::Instance Utilice la propiedad <code>CreditSpecification</code> para especificar la opción de crédito para el uso de CPU de una instancia T2. Utilice la propiedad <code>ElasticGpuSpecifications</code> para especificar GPU elásticas, que son recursos de GPU que se pueden adjuntar a la instancia para acelerar el desempeño gráfico de las aplicaciones. AWS::EC2::VPC La propiedad <code>InstanceTenancy</code> ya no requiere una interrupción cuando se actualiza de "dedicated" a "default" . AWS::ECS::Service Utilice la propiedad <code>HealthCheckGracePeriodSeconds</code> para especificar el periodo de tiempo, en segundos, durante el cual el programador de servicio de Amazon ECS debe hacer caso omiso de las comprobaciones de estado de los destinos de Elastic Load Balancing en mal estado después de que se haya iniciado una tarea por primera vez.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
		AWS::IoT::TopicRule En el tipo de propiedad DynamoDBAction, las propiedades <code>RangeKeyField</code> y <code>RangeKeyValue</code> ya no son necesarias. AWS::KinesisAnalytics::ApplicationOutput En el tipo de propiedad ApplicationOutput, utilice la propiedad <code>LambdaOutput</code> para identificar una función de Lambda como destino al configurar la salida de la aplicación. AWS::Kinesis::Stream Utilice la propiedad <code>StreamEncryption</code> para habilitar o actualizar el cifrado del servidor con una AWS KMS key para un flujo determinado. AWS::Lambda::Function Utilice la propiedad <code>ReservedConcurrentExecutions</code> para especificar el número máximo de ejecuciones simultáneas que desea reservar para la función. AWS::RDS::DBSubnetGroup Utilice la propiedad <code>DBSubnetGroupName</code> para especificar el nombre del grupo de subredes de base de datos. AWS::S3::Bucket Utilice la propiedad <code>BucketEncryption</code> para especificar el cifrado predeterminado para un bucket utilizando el cifrado del lado del servidor con claves administradas por Amazon S3 (SSE-S3) o un bucket de AWS KMS keys (SSE-KMS).	

Cambio	Fecha de lanzamiento	Descripción	Versión de API
		En el tipo de propiedad ReplicationRule, utilice la propiedad <code>SourceSelectionCriteria</code> para especificar filtros adicionales para la identificación de los objetos de origen que desea replicar. En el tipo de propiedad ReplicationDestination: • Utilice la propiedad <code>AccessControlTranslation</code> para especificar la titularidad de la réplica de la cuenta de AWS a la que pertenece el bucket de destino. • Utilice la propiedad <code>Account</code> para especificar el ID de cuenta del propietario del bucket de destino. • Utilice la propiedad <code>EncryptionConfiguration</code> para especificar información relacionada con el cifrado de un bucket que es un destino para los objetos replicados. AWS::SSM::Association Utilice la propiedad <code>AssociationName</code> para especificar el nombre de la asociación entre un documento de SSM y las instancias de EC2 que contienen un agente de configuración para procesar el documento.	
Se han añadido desencadenadores de restauración a la consola de CloudFormation.	15 de enero de 2018	Los desencadenadores de restauración permiten que CloudFormation supervise el estado de su aplicación durante la creación y la actualización de la pila y para restaurar dicha operación si la aplicación sobrepasa el umbral de cualquiera de las alarmas especificadas. Para obtener más información, consulte Monitorización y reversión de las operaciones de la pila.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Recurso actualizado	12 de enero de 2018	AWS::SSM::Parameter Utilice la propiedad <code>AllowedPattern</code> para especificar una expresión regular utilizada para validar el valor del parámetro.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Recursos nuevos	5 de diciembre de 2017	AWS::Inspector::AssessmentTarget Use el recurso <code>AWS::Inspector::AssessmentTarget</code> para crear un objetivo de evaluación de Amazon Inspector. AWS::Inspector::AssessmentTemplate Use el recurso <code>AWS::Inspector::AssessmentTemplate</code> para crear una plantilla de evaluación de Amazon Inspector. AWS::Inspector::ResourceGroup Utilice el recurso <code>AWS::Inspector::ResourceGroup</code> para crear un grupo de recursos de Amazon Inspector, que define las etiquetas para identificar los recursos de AWS que forman un objetivo de evaluación de Amazon Inspector. AWS::ServiceDiscovery::Instance Utilice el recurso <code>AWS::ServiceDiscovery::Instance</code> para especificar información sobre una instancia que crea Amazon Route 53. AWS::ServiceDiscovery::PrivateDnsNamespace Utilice el recurso <code>AWS::ServiceDiscovery::PrivateDnsNamespace</code> para especificar información sobre un espacio de nombres privado para Amazon Route 53. AWS::ServiceDiscovery::PublicDnsNamespace Utilice el recurso <code>AWS::ServiceDiscovery::PublicDnsNamespace</code> para especificar	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
		información sobre un espacio de nombres público para Amazon Route 53. AWS::ServiceDiscovery::Service Utilice el recurso <code>AWS::ServiceDiscovery::Service</code> para definir una plantilla para un máximo de cinco registros y una comprobación de estado opcional que desea que cree Amazon Route 53 al registrar una instancia.	
Recurso actualizado	5 de diciembre de 2017	AWS::KinesisAnalytics::Application En el tipo de propiedad Input, utilice la propiedad <code>InputProcessingConfiguration</code> para transformar registros a medida que se reciben desde la secuencia.	2010-05-15
Recurso actualizado	1 de diciembre de 2017	AWS::CodeBuild::Project Utilice la propiedad <code>BadgeEnabled</code> para generar una URL de acceso público para la insignia de compilación de un proyecto. Utilice la propiedad <code>Cache</code> para configurar la configuración de caché para crear dependencias de compilación. Utilice la propiedad <code>VpcConfig</code> para permitir que CodeBuild acceda a recursos en una VPC de Amazon. En el tipo de propiedad EnvironmentVariable, utilice la propiedad <code>Type</code> para especificar el tipo de variable de entorno.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Recurso nuevo	30 de noviembre de 2017	AWS::Cloud9::EnvironmentEC2 Utilice el recurso <code>AWS::Cloud9::EnvironmentEC2</code> para crear un entorno de desarrollo de Amazon EC2 en AWS Cloud9.	2010-05-15
Recursos actualizados	29 de noviembre de 2017	AWS::ECS::TaskDefinition Utilice la propiedad <code>Cpu</code> para especificar el número de unidades de CPU necesarias para la tarea. Utilice la propiedad <code>ExecutionRoleArn</code> para especificar el ARN del rol de ejecución. Utilice la propiedad <code>Memory</code> para especificar la cantidad de memoria (en MiB) necesaria para la tarea. Utilice la propiedad <code>RequiresCompatibilities</code> para especificar el tipo de lanzamiento que requiere la tarea. AWS::ECS::Service Utilice la propiedad <code>LaunchType</code> para especificar el tipo de lanzamiento en el que se va a ejecutar el servicio. Utilice la propiedad <code>NetworkConfiguration</code> para especificar la configuración de red para el servicio. Utilice la propiedad <code>PlatformVersion</code> para especificar la versión de la plataforma en la que se va a ejecutar el servicio.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Recursos nuevos	28 de noviembre de 2017	AWS::GuardDuty::Detector Utilice el recurso <code>AWS::GuardDuty::Detector</code> para crear un detector de Amazon GuardDuty único. AWS::GuardDuty::IPSet Use el recurso <code>AWS::GuardDuty::IPSet</code> para crear un conjunto de IP de Amazon GuardDuty. AWS::GuardDuty::ThreatIntelSet Utilice el recurso <code>AWS::GuardDuty::ThreatIntelSet</code> para crear un ThreatIntelSet.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Recursos actualizados	28 de noviembre de 2017	AWS::CodeDeploy::Application Utilice la propiedad <code>ComputePlatform</code> para especificar una plataforma de computación de AWS Lambda en la que CodeDeploy va a implementar una aplicación. AWS::CodeDeploy::DeploymentGroup En el tipo de propiedad DeploymentStyle, utilice la propiedad <code>DeploymentType</code> para especificar una implementación azul/verde en una plataforma de computación de Lambda. AWS::EC2::SpotFleet En el tipo de propiedad SpotFleetRequestConfigData, ahora la propiedad <code>SpotPrice</code> es opcional. AWS::Lambda::Alias Utilice la propiedad <code>RoutingConfig</code> para especificar dos versiones diferentes de una función AWS Lambda, lo que le permite dictar qué porcentaje de tráfico invocará cada versión.	2010-05-15
Nueva política de actualización de CodeDeployLambdaAliasUpdate	28 de noviembre de 2017	Use la política de actualización <code>CodeDeployLambdaAliasUpdate</code> para realizar una implementación de CodeDeploy cuando la versión cambia en un recurso de <code>AWS::Lambda::Alias</code> . Para obtener más información, consulte Atributo UpdatePolicy .	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Nuevos tipos de parámetros SSM	21 de noviembre de 2017	Utilice los tipos de parámetros SSM para utilizar los parámetros existentes de Administrador de sistemas Parameter Store. Nota: CloudFormation no admite actualmente el tipo <code>SecureString</code> . Para obtener más información, consulte Tipos de parámetros SSM .	2010-05-15
Nuevo campo <code>ResolvedValue</code> para el tipo de dato <code>Parameter</code>	21 de noviembre de 2017	El campo <code>ResolvedValue</code> devuelve el valor que se utiliza en la definición de pila de un parámetro SSM. Para obtener más información, consulte el tipo de datos Parameter en Referencia de la API de CloudFormation.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Recursos actualizados	20 de noviembre de 2017	AWS::ApiGateway::ApiKey Utilice la propiedad <code>CustomerId</code> para especificar un identificador de cliente de AWS Marketplace. Utilice la propiedad <code>GenerateDistinctId</code> para especificar si el identificador de la clave es diferente del valor de clave de API creado. AWS::ApiGateway::Authorizer Utilice la propiedad <code>AuthType</code> para especificar un campo definido por el cliente que se utiliza en importaciones y exportaciones de Swagger sin impacto funcional. AWS::ApiGateway::DomainName Utilice la propiedad <code>EndpointConfiguration</code> para especificar los tipos de punto de enlace de un nombre de dominio de API Gateway. Utilice la propiedad <code>RegionalCertificateArn</code> para hacer referencia a un certificado que utiliza el punto de enlace regional para un nombre de dominio. AWS::ApiGateway::Method En los tipos de propiedad Integration y IntegrationResponse, utilice la propiedad <code>ContentHandling</code> para especificar cómo gestionar las conversiones de tipos de contenido de carga útil de las solicitudes.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
		AWS::ApiGateway::RestApi Utilice la propiedad <code>EndpointConfiguration</code> para especificar los tipos de punto de enlace de una API REST de API Gateway. AWS::ApplicationAutoScaling::ScalableTarget Utilice la propiedad <code>ScheduledActions</code> para especificar acciones programadas para un destino escalable de Application Auto Scaling. AWS::ECR::Repository Utilice la propiedad <code>LifecyclePolicy</code> para especificar una política de ciclo de vida para un repositorio de Amazon ECR. AWS::ECS::TaskDefinition En el tipo de propiedad ContainerDefinition, utilice la propiedad <code>LinuxParameters</code> para especificar las opciones específicas de Linux para un contenedor de Amazon ECS. AWS::ElastiCache::ReplicationGroup Utilice la propiedad <code>AtRestEncryptionEnabled</code> para habilitar el cifrado en reposo. Utilice la propiedad <code>AuthToken</code> para especificar la contraseña que se utiliza para acceder a un servidor protegido con contraseña. Utilice la propiedad <code>TransitEncryptionEnabled</code> para habilitar el cifrado en tránsito.	

Cambio	Fecha de lanzamiento	Descripción	Versión de API
		AWS::ElasticLoadBalancingV2::TargetGroup Utilice el atributo <code>TargetGroupName</code> con la función <code>Fn::GetAtt</code> para obtener el nombre de un grupo de destino de Elastic Load Balancing. AWS::Elasticsearch::Domain Utilice la propiedad <code>VPCOptions</code> para especificar una configuración de VPC para el dominio de OpenSearch Service. AWS::EMR::Cluster Utilice la propiedad <code>EbsRootVolumeSize</code> para especificar el tamaño del volumen raíz de EBS para un clúster de Amazon EMR. AWS::RDS::DBInstance Utilice las propiedades <code>SourceRegion</code> y <code>KmsKeyId</code> para crear una réplica de lectura cifrada a partir de una instancia de base de datos de origen entre regiones. AWS::Route53::HostedZone Utilice la propiedad <code>QueryLoggingConfig</code> para especificar una configuración para el registro de consultas de DNS.	

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Nuevo campo NoEcho para objetos Response de recursos personalizados	20 de noviembre de 2017	A partir de ahora, puede utilizar el campo NoEcho opcional para enmascarar la salida de un recurso personalizado. Para obtener más información, consulte Objetos de respuesta de recursos personalizados. El parámetro noEcho correspondiente es compatible con el método send. Para obtener más información, consulte Módulo cfn-response.	2010-05-15
Se agregaron anulaciones de instancias de pila para los conjuntos de pilas.	17 de noviembre de 2017	CloudFormation StackSets permite anular los valores de los parámetros en instancias de pila por cuenta y región. Puede anular los valores de los parámetros al crear las instancias de pila o al actualizar instancias de pila existentes. Para obtener más información, consulte Anulación de parámetros en instancias de pila .	2010-05-15
Recurso actualizado	15 de noviembre de 2017	AWS::StepFunctions::StateMachine Puede utilizar <code>AWS::StepFunctions::StateMachine</code> para especificar un <code>StateMachineName</code> al crear una máquina de estado y puede actualizar <code>DefinitionString</code> y <code>RoleArn</code> sin sustituir la máquina de estado.	2010-05-15
StackSets ahora admite un máximo de 500 instancias de pila por conjunto de pilas.	6 de noviembre de 2017	Ahora puede crear hasta un máximo de 500 instancias de pila por conjunto de pilas. Para obtener más información sobre los límites de CloudFormation, consulte Comprensión de las cuotas de CloudFormation .	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Recursos nuevos	2 de noviembre de 2017	AWS::CloudFront::CloudFrontOriginAccessIdentity Utilice el recurso <code>AWS::CloudFront::CloudFrontOriginAccessIdentity</code> para especificar la identidad de acceso de origen de Amazon CloudFront que debe asociarse al origen de una distribución de CloudFront. AWS::CloudFront::StreamingDistribution Utilice el recurso <code>AWS::CloudFront::StreamingDistribution</code> para especificar una distribución de streaming del Protocolo de mensajería en tiempo real (RTMP) de Adobe para CloudFront.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Recursos actualizados	2 de noviembre de 2017	AWS::ApiGateway::Deployment La propiedad <code>StageName</code> no está disponible en el tipo de propiedad StageDescription. AWS::ApiGateway::Method Utilice la propiedad <code>OperationName</code> para asignar un nombre fácil de recordar a un método de API Gateway. Utilice la propiedad <code>RequestValidatorId</code> para asociar un validador de solicitud a un método. AWS::AutoScaling::AutoScalingGroup Utilice la propiedad <code>LifecycleHookSpecificationList</code> para especificar las acciones que se realizarán cuando Auto Scaling lance o termine instancias. AWS::CloudFront::Distribution Utilice la propiedad <code>Tags</code> para especificar un conjunto arbitrario de etiquetas (pares clave-valor) para asociar a una distribución de CloudFront. En los tipos de propiedad CacheBehavior y DefaultCacheBehavior, utilice la propiedad <code>LambdaFunctionAssociations</code> para especificar las asociaciones de función de Lambda para una distribución de CloudFront. En el tipo de propiedad CustomOriginConfig, utilice la propiedad <code>OriginKeepaliveTimeout</code> para especificar un tiempo de espera keep-alive personalizado y utilice la propiedad <code>OriginReadTimeout</code>	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
		para especificar un tiempo de espera de lectura de origen personalizado. En el tipo de propiedad DistributionConfig, utilice la propiedad <code>IPV6Enabled</code> para especificar si CloudFront responde a solicitudes DNS IPv6 con una dirección IPv6 para la distribución. AWS::CodeDeploy::DeploymentGroup En el tipo de propiedad LoadBalancerInfo, utilice la propiedad <code>TargetGroupInfoList</code> para especificar información sobre un grupo de destino en Elastic Load Balancing que se utilizará en una implementación. AWS::EC2::SecurityGroup, AWS::EC2::SecurityGroupEgress, y AWS::EC2::SecurityGroupIngress Utilice la propiedad <code>Description</code> para especificar la descripción de una regla de grupo de seguridad. AWS::EC2::Subnet La propiedad <code>Ipv6CidrBlock</code> es ahora compatible con las actualizaciones de <code>No interruption</code>. AWS::EC2::VPNGateway Utilice la propiedad <code>AmazonSideAsn</code> para especificar un número de sistema autónomo privado (ASN) del lado de Amazon de una sesión BGP. AWS::EC2::VPNConnection Utilice la propiedad <code>VpnTunnelOptionsSpecifications</code> para configurar las opciones de túnel de una conexión de VPN.	

Cambio	Fecha de lanzamiento	Descripción	Versión de API
		AWS::ElasticBeanstalk::ConfigurationTemplate y AWS::ElasticBeanstalk::Environment En los tipos de propiedad ConfigurationOptionSetting y OptionSetting, utilice la propiedad ResourceName para especificar un nombre de recurso para una opción de configuración de escalado basada en tiempo. AWS::EMR::Cluster Utilice la propiedad CustomAmiId para especificar una AMI de Amazon Linux personalizada para un clúster. AWS::KinesisFirehose::DeliveryStream Utilice el atributo Arn con la función Fn::GetAtt para obtener el nombre de recurso de Amazon (ARN) del flujo de entrega. AWS::KMS::Key Utilice la propiedad Tags para especificar un conjunto arbitrario de etiquetas (pares clave-valor) para asociar a una clave administrada por el cliente. AWS::OpsWorks::Layer y AWS::OpsWorks::Stack Utilice la propiedad Tags para especificar un conjunto arbitrario de etiquetas (pares clave-valor) para asociar a una capa o pila de OpsWorks. AWS::RDS::OptionGroup En el tipo de propiedad OptionConfiguration, utilice la propiedad OptionVersion para especificar una versión de la opción.	

Cambio	Fecha de lanzamiento	Descripción	Versión de API
		AWS::S3::Bucket Utilice la propiedad <code>AnalyticsConfigurations</code> para configurar un filtro de análisis para un bucket de Amazon S3.	

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Recursos nuevos	24 de octubre de 2017	AWS::Glue::Classifier Utilice el recurso <code>AWS::Glue::Classifier</code> para crear un clasificador de AWS Glue. AWS::Glue::Connection Utilice el recurso <code>AWS::Glue::Connection</code> para especificar una conexión de AWS Glue a un origen de datos. AWS::Glue::Crawler Utilice el recurso <code>AWS::Glue::Crawler</code> para especificar un rastreador de AWS Glue. AWS::Glue::Database Utilice el recurso <code>AWS::Glue::Database</code> para crear una base de datos de AWS Glue. AWS::Glue::DevEndpoint Utilice el recurso <code>AWS::Glue::DevEndpoint</code> para especificar un punto de enlace de desarrollo para la depuración remota de los scripts ETL. AWS::Glue::Job Utilice el recurso <code>AWS::Glue::Job</code> para especificar una tarea de AWS Glue en el catálogo de datos. AWS::Glue::Partition Utilice el recurso <code>AWS::Glue::Partition</code> para crear una partición de AWS Glue, que representa una parte de los datos de la tabla.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
		AWS::Glue::Table Utilice el recurso <code>AWS::Glue::Table</code> para crear una tabla de AWS Glue. AWS::Glue::Trigger Utilice el recurso <code>AWS::Glue::Trigger</code> para especificar disparadores que ejecutan trabajos de AWS Glue.	
Recursos nuevos	11 de octubre de 2017	AWS::SSM::MaintenanceWindow Utilice el recurso <code>AWS::SSM::MaintenanceWindow</code> para crear un periodo de mantenimiento de AWS Systems Manager. AWS::SSM::MaintenanceWindowTarget Utilice el recurso <code>AWS::SSM::MaintenanceWindowTarget</code> para registrar un destino con un periodo de mantenimiento. AWS::SSM::MaintenanceWindowTask Utilice el recurso <code>AWS::SSM::MaintenanceWindowTask</code> para definir una tarea de periodo de mantenimiento. AWS::SSM::PatchBaseline Utilice el recurso <code>AWS::SSM::PatchBaseline</code> para definir una base de referencia de parches de Administrador de sistemas.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Recurso nuevo	10 de octubre de 2017	AWS::ElasticLoadBalancingV2::ListenerCertificate Utilice el recurso <code>AWS::ElasticLoadBalancingV2::ListenerCertificate</code> para especificar certificados para un agente de escucha de Elastic Load Balancing.	2010-05-15
Recurso nuevo	27 de septiembre de 2017	AWS::Athena::NamedQuery Utilice el recurso <code>AWS::Athena::NamedQuery</code> para crear una consulta de Amazon Athena.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Recursos actualizados	27 de septiembre de 2017	AWS::EC2::NatGateway Utilice la propiedad <code>Tags</code> para especificar las etiquetas de recursos para una puerta de enlace NAT. AWS::ElasticBeanstalk::Application Utilice la propiedad <code>ResourceLifecycleConfig</code> para definir la configuración del ciclo de vida de recursos que pertenecen a la aplicación y el rol de servicio que Elastic Beanstalk asume para aplicar la configuración del ciclo de vida. AWS::ElasticBeanstalk::ConfigurationTemplate y AWS::ElasticBeanstalk::Environment Utilice la propiedad <code>PlatformArn</code> para especificar una plataforma personalizada para Elastic Beanstalk. AWS::ElasticLoadBalancingV2::TargetGroup En el tipo de propiedad TargetDescription, utilice la propiedad <code>AvailabilityZone</code> para especificar la zona de disponibilidad en la que se registrará la dirección IP. AWS::Events::Rule En el tipo de propiedad Target, utilice las siguientes propiedades de transformación de entrada de eventos y configuración de tareas de Amazon ECS y destinos de flujos de Kinesis. • <code>EcsParameters</code> • <code>InputTransformer</code> • <code>KinesisParameters</code>	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
		RunCommandParameters AWS::KinesisFirehose::DeliveryStream Utilice la propiedad <code>DeliveryStreamType</code> para especificar el tipo de flujo y la propiedad <code>KinesisStreamSourceConfiguration</code> para especificar el flujo y los roles de ARN de un flujo de Kinesis utilizado como origen de un flujo de entrega. AWS::RDS::DBInstance En el caso de la propiedad <code>Engine</code>, si ha especificado <code>oracle-se</code> u <code>oracle-se1</code>, puede actualizar a <code>oracle-se2</code> sin necesidad de sustitución de la instancia de la base de datos. AWS::S3::Bucket Utilice la propiedad <code>AccelerateConfiguration</code> para configurar el estado aceleración de transferencia de un bucket de Amazon S3.	
Se agregó protección de terminación para pilas.	26 de septiembre de 2017	Habilitar la protección de terminación en una pila impide que se elimine accidentalmente. Un usuario no puede eliminar una pila con protección de terminación habilitada. Para obtener más información, consulte Proteger una pila para impedir que la eliminen accidentalmente.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Se ha cambiado el valor predeterminado de umask de la versión 1.4-22 y posteriores	14 de septiembre de 2017	El valor predeterminado del parámetro umask del archivo de configuración cfn-hup.conf ahora es 022. Para obtener más información, consulte cfn-hup .	
Recursos actualizados	7 de septiembre de 2017	AWS::ElasticLoadBalancingV2::LoadBalancer Utilice la propiedad SubnetMappings para especificar los ID de las subredes que adjuntar al balanceador de carga. Utilice la propiedad Type para especificar el tipo de balanceador de carga que crear. AWS::ElasticLoadBalancingV2::TargetGroup Utilice la propiedad TargetType para especificar el tipo de registro de los objetivos de este grupo de destino.	2010-05-15
Desencadenadores de restauración añadidos a la API de CloudFormation	31 de agosto de 2017	Los desencadenadores de restauración permiten que CloudFormation supervise el estado de su aplicación durante la creación y la actualización de la pila y para restaurar dicha operación si la aplicación sobrepasa el umbral de cualquiera de las alarmas especificadas. Para obtener más información, consulte RollbackConfiguration en la Referencia de la API de CloudFormation.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Nuevo parámetro <code>umask</code> para el archivo <code>cfn-hup.conf</code>	31 de agosto de 2017	Utilice el parámetro <code>umask</code> del archivo de configuración <code>cfn-hup.conf</code> para controlar los permisos de archivos utilizados por el daemon <code>cfn-hup</code> (versión 1.4-21). Para obtener más información, consulte cfn-hup .	
Recursos actualizados para compatibilidad con tamaño de VPC	29 de agosto de 2017	AWS::EC2::VPC::CidrBlock Utilice la propiedad <code>CidrBlock</code> para asociar un bloque de CIDR IPv4 con una VPC. AWS::EC2::VPC Utilice el atributo <code>CidrBlockAssociations</code> con la función <code>Fn::GetAtt</code> para obtener una lista de IDs de asociación de bloques de CIDR IPv4 asociada a la VPC.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Recursos actualizados	23 de agosto de 2017	AWS::S3::Bucket En el tipo de propiedad Rule, utilice la propiedad <code>TagFilters</code> para especificar etiquetas que utilizar en la identificación de un subconjunto de objetos para un bucket de Amazon S3. Utilice la propiedad <code>MetricsConfiguration</code> para especificar una configuración de las métricas para las métricas de solicitudes de CloudWatch desde un bucket de Amazon S3. AWS::IoT::TopicRule En el tipo de propiedad Action, utilice la propiedad <code>DynamoDBv2Action</code> para describir una acción de AWS IoT que escriba datos en una tabla de DynamoDB. En el tipo de propiedad Action, la propiedad <code>DynamoDBAction</code> ahora es compatible con las propiedades <code>HashKeyType</code> y <code>RangeKeyType</code>. AWS::Lambda::Permission Utilice la propiedad <code>EventSourceToken</code> para especificar un token único que debe suministrarla el principal que invoca la función.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Nuevos pseudoparámetros	23 de agosto de 2017	Utilice el pseudoparámetro <code>AWS::Partition</code> para devolver la partición en la que se encuentra un recurso. Utilice el pseudoparámetro <code>AWS::URLSuffix</code> para devolver el sufijo de un dominio. Para obtener más información, consulte Referencia de pseudoparámetros.	2010-05-15
Nuevos recursos para el soporte de DAX	22 de agosto de 2017	AWS::DAX::Cluster Utilice el recurso <code>AWS::DAX::Cluster</code> para crear un clúster de DAX para utilizarlo con Amazon DynamoDB. AWS::DAX::ParameterGroup Utilice el recurso <code>AWS::DAX::ParameterGroup</code> para crear un grupo de parámetros para utilizarlo con Amazon DynamoDB. AWS::DAX::SubnetGroup Utilice el recurso <code>AWS::DAX::SubnetGroup</code> para crear un grupo de parámetros para utilizarlo con DAX (DynamoDB Accelerator).	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Recursos nuevos	18 de agosto de 2017	AWS::ApiGateway::DocumentationPart y AWS::ApiGateway::DocumentationVersion para crear documentación para su API de API Gateway. AWS::ApiGateway::GatewayResponse Utilice el recurso <code>AWS::ApiGateway::GatewayResponse</code> para crear una respuesta personalizada para su API de API Gateway. AWS::ApiGateway::RequestValidator Utilice el recurso <code>AWS::ApiGateway::RequestValidator</code> para configurar reglas de validación de las solicitudes entrantes a la API de API Gateway. AWS::EC2::NetworkInterfacePermission Utilice el recurso <code>AWS::EC2::NetworkInterfacePermission</code> para otorgar permisos de cuenta de AWS a una interfaz de red.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Recursos actualizados	18 de agosto de 2017	AWS::ApiGateway::Stage Utilice la propiedad <code>DocumentationVersion</code> para especificar una instantánea con control de versiones de la documentación de la API. AWS::AutoScaling::ScalingPolicy Utilice la propiedad <code>TargetTrackingConfiguration</code> para especificar una configuración de la política de escalado de seguimiento de destino de Auto Scaling. AWS::CloudTrail::Trail Utilice la propiedad <code>EventSelectors</code> para prestar soporte de eventos de datos de Amazon S3. AWS::CodeDeploy::DeploymentGroup Utilice el las propiedades <code>LoadBalancerInfo</code> y <code>DeploymentStyle</code> para especificar un balanceador de carga de Elastic Load Balancing para una implementación in situ. Utilice la propiedad <code>AutoRollbackConfiguration</code> para configurar la restauración automática para la implementación. AWS::EC2::SpotFleet En el tipo de propiedad SpotFleetRequestConfigData, utilice la propiedad <code>ReplaceUnhealthyInstances</code> para indicar si la flota de spot debería sustituir instancias en mal estado y la propiedad <code>Type</code> para especificar el tipo de solicitud.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
		AWS::EC2::Subnet Utilice las propiedades <code>AssignIpv6AddressOnCreation</code> y <code>Ipv6CidrBlock</code> para crear una subred con un bloque de CIDR IPv6. AWS::KinesisFirehose::DeliveryStream Utilice la propiedad <code>ExtendedS3DestinationConfiguration</code> para configurar un destino en Amazon S3. Utilice la subpropiedad <code>ProcessingConfiguration</code> dentro de cada configuración de destino para invocar funciones de Lambda que transforman los datos de la fuente entrante y entregan los datos transformados en los destinos. AWS::RDS::DBCluster y AWS::RDS::DBInstance La <code>DeletionPolicy</code> predeterminada es ahora <code>Snapshot</code> para recursos <code>AWS::RDS::DBCluster</code> y para recursos <code>AWS::RDS::DBInstance</code> que no especifican la propiedad <code>DBClusterIdentifier</code>. Para obtener más información, consulte Atributo DeletionPolicy. AWS::S3::Bucket En el tipo de propiedad Rule, utilice la propiedad <code>AbortIncompleteMultipartUpload</code> para especificar una regla de ciclo de vida que anula las cargas multiparte incompletas a un bucket de Amazon S3.	

Cambio	Fecha de lanzamiento	Descripción	Versión de API
		AWS::SQS::Queue Utilice las propiedades <code>KmsMasterKeyId</code> y <code>KmsDataKeyReusePeriodSeconds</code> para configurar el cifrado del servidor para Amazon SQS. Se agregó el atributo <code>Arn</code> a la función intrínseca <code>Fn::GetAtt</code> para los siguientes recursos: • AWS::CloudTrail::Trail. También se agregó <code>SnsTopicArn</code>. • AWS::CloudWatch::Alarm • AWS::DynamoDB::Table • AWS::ECS::Cluster • AWS::IoT::Policy • AWS::IoT::TopicRule • AWS::Logs::Destination	
Soporte para etiquetas de pila en elementos de CodePipeline	18 de agosto de 2017	Ahora puede especificar etiquetas para pilas en archivos de configuración de plantillas para usarlos como elementos para canalizaciones de CodePipeline. Las etiquetas especificadas se aplican a las pilas creadas con el archivo de configuración de plantilla. Para obtener más información, consulte Artefactos de CloudFormation .	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Crear sistemas de archivos cifrados	14 de agosto de 2017	AWS::EFS::FileSystem Utilice la propiedad <code>Encrypted</code> para cifrar un sistema de archivos Amazon EFS durante la creación. Utilice la propiedad <code>KmsKeyId</code> para especificar opcionalmente una clave administrada por el cliente personalizada que utilizar para proteger el sistema de archivos cifrados.	2010-05-15
Nuevos recursos para soporte de AWS Batch	8 de agosto de 2017	AWS::Batch::ComputeEnvironment Utilice el recurso <code>AWS::Batch::ComputeEnvironment</code> para definir el entorno informático de AWS Batch. AWS::Batch::JobDefinition Utilice el recurso <code>AWS::Batch::JobDefinition</code> para especificar los parámetros para una definición de trabajo de AWS Batch. AWS::Batch::JobQueue Utilice el recurso <code>AWS::Batch::JobQueue</code> para definir la cola de trabajos de AWS Batch.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Nuevos recursos de compatibilidad de Amazon Managed Service para Apache Flink	28 de julio de 2017	AWS::KinesisAnalytics::Application Use el recurso <code>AWS::KinesisAnalytics::Application</code> para crear una aplicación de Amazon Managed Service para Apache Flink. AWS::KinesisAnalytics::ApplicationOutput Use el recurso <code>AWS::KinesisAnalytics::ApplicationOutput</code> para agregar un destino externo a su aplicación de Amazon Managed Service para Apache Flink. AWS::KinesisAnalytics::ApplicationReferenceDataSource Use el recurso <code>AWS::KinesisAnalytics::ApplicationReferenceDataSource</code> para agregar un origen de datos de referencia a una aplicación existente de Amazon Managed Service para Apache Flink.	2010-05-15
Utilice StackSets para administrar de forma centralizada las pilas en cuentas y regiones	25 de julio de 2017	StackSets permite crear, actualizar o eliminar pilas de varias cuentas y regiones en una sola operación. Con una cuenta de administrador, defina y administre una plantilla de CloudFormation y utilice esta plantilla como base para aprovisionar pilas en cuentas de destino seleccionadas a lo largo de las regiones especificadas. Para obtener más información, consulte Administración de pilas entre cuentas y regiones con StackSets.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Ver eventos de la pila por token de solicitud de cliente	14 de julio de 2017	En la consola, las operaciones de pila muestran el token de solicitud del cliente de la pestaña Events (Eventos) . A todos los eventos activados por una determinada operación de la pila se les asigna el mismo token de solicitud de cliente, que puede utilizar para hacer un seguimiento de las operaciones. Para obtener más información, consulte Visualización de los recursos y datos de la pila de CloudFormation en la Consola de administración de AWS y StackEvent y en la Referencia de la API de AWS CloudFormation.	2010-05-15
Utilizar enlaces Quick-Create de pila	14 de julio de 2017	Utilice los enlaces Quick-Create para poner en funcionamiento pilas rápidamente. Puede especificar el URL de la plantilla, el nombre de la pila y los parámetros de la plantilla para rellenar una sola página del Create Stack Wizard (Asistente para creación de pila). Para obtener más información, consulte Creación de enlaces Quick-Create para pilas .	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Nuevos recursos para soporte de AWS Database Migration Service	12 de julio de 2017	AWS::DMS::Certificate Utilice el recurso <code>AWS::DMS::Certificate</code> para crear un certificado SSL que cifra las conexiones entre los puntos de enlace de AWS DMS y la instancia de replicación. AWS::DMS::Endpoint Use el recurso <code>AWS::DMS::Endpoint</code> para crear un punto de enlace de AWS DMS. AWS::DMS::EventSubscription Utilice el recurso <code>AWS::DMS::EventSubscription</code> para recibir notificaciones para eventos de AWS DMS a través de Amazon Simple Notification Service. AWS::DMS::ReplicationInstance Utilice el recurso <code>AWS::DMS::ReplicationInstance</code> para crear una instancia de replicación de AWS DMS. AWS::DMS::ReplicationSubnetGroup Utilice el recurso <code>AWS::DMS::ReplicationSubnetGroup</code> para crear un grupo de subred de replicación AWS DMS. AWS::DMS::ReplicationTask Utilice el recurso <code>AWS::DMS::ReplicationTask</code> para crear una tarea de replicación AWS DMS.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Recursos nuevos	5 de julio de 2017	AWS::CloudWatch::Dashboard Utilice el recurso <code>AWS::CloudWatch::Dashboard</code> para especificar un panel de CloudWatch personalizado para su consola de CloudWatch. AWS::ApiGateway::DomainName Utilice el recurso <code>AWS::ApiGateway::DomainName</code> para especificar una URL personalizada e intuitiva para la API implementada en Amazon API Gateway. AWS::EC2::EgressOnlyInternetGateway Utilice el recurso <code>AWS::EC2::EgressOnlyInternetGateway</code> para crear un puerto de enlace a Internet de solo salida para su VPC. InstanceFleetConfig Utilice el recurso <code>InstanceFleetConfig</code> para configurar una flota de instancias de spot para un clúster de Amazon EMR.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Recursos actualizados	5 de julio de 2017	AWS::ApiGateway::RestApi Utilice la propiedad <code>BinaryMediaTypes</code> para especificar los tipos de medios binarios compatibles. AWS::ApplicationAutoScaling::ScalingPolicy Utilice la propiedad <code>TargetTrackingScalingPolicyConfiguration</code> para especificar una configuración de la política de escalado de seguimiento de destino. AWS::CloudTrail::Trail Utilice la propiedad <code>TrailName</code> para especificar un nombre personalizado para un recurso de AWS CloudTrail. Utilice la propiedad <code>Tags</code> para especificar etiquetas de recursos. AWS::CodeDeploy::DeploymentGroup Utilice la propiedad <code>AlarmConfiguration</code> para configurar alarmas para el grupo de implementación. Utilice la propiedad <code>TriggerConfigurations</code> para configurar activadores de notificaciones para el grupo de implementación. AWS::EMR::Cluster Utilice la propiedad <code>CoreInstanceFleet</code> y la propiedad <code>MasterInstanceFleet</code> en el tipo de propiedad JobFlowInstancesConfig para configurar la flota de instancias de spot para un clúster de Amazon EMR.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
		AWS::DynamoDB::Table Utilice la propiedad <code>TimeToLiveSpecification</code> para especificar la configuración del periodo de vida (TTL) para una tabla de Amazon DynamoDB. Utilice la propiedad <code>Tags</code> para especificar las etiquetas de recursos para una tabla de DynamoDB. AWS::EC2::Instance La propiedad <code>IamInstanceProfile</code> es ahora compatible con las actualizaciones de <code>No interruption</code>. AWS::EC2::Route Utilice la propiedad <code>EgressOnlyInternetGatewayId</code> para especificar un puerto de enlace a Internet de solo salida para una ruta de EC2. AWS::Kinesis::Stream Utilice la propiedad <code>RetentionPeriodHours</code> para especificar el número de horas que los registros de datos almacenados en fragmentos permanecen accesibles. AWS::RDS::DBCluster Utilice la propiedad <code>ReplicationSourceIdentifier</code> para crear un clúster de base de datos como réplica de lectura de otro clúster de base de datos o una instancia de base de datos MySQL de Amazon RDS.	

Cambio	Fecha de lanzamiento	Descripción	Versión de API
		AWS::Redshift::Cluster Utilice la propiedad <code>LoggingProperties</code> para crear archivos de registro de auditoría y almacenarlos en Amazon S3.	
Recursos nuevos	6 de junio de 2017	AWS::EMR::SecurityConfiguration Utilice el recurso <code>AWS::EMR::SecurityConfiguration</code> para crear una configuración de seguridad que se almacena en el servicio y puede especificarse cuando se crea un clúster.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Recursos actualizados	6 de junio de 2017	AWS::AutoScaling::LifecycleHook Las propiedades <code>NotificationTargetARN</code> y <code>RoleARN</code> son ahora opcionales. AWS::CloudWatch::Alarm Ahora puede utilizar las propiedades <code>EvaluateLowSampleCountPercentile</code>, <code>ExtendedStatistic</code> y <code>TreatMissingData</code> al crear recursos de <code>AWS::CloudWatch::Alarm</code>. AWS::EC2::SpotFleet CloudFormation admite cambios mutables en las propiedades de flota de spot. Las propiedades siguientes de la propiedad <code>SpotFleetRequestConfigData</code> admiten actualizaciones de <code>Replacement</code>: • <code>AllocationStrategy</code> • <code>IamFleetRole</code> • <code>LaunchSpecifications</code> • <code>SpotPrice</code> • <code>TerminateInstancesWithExpiration</code> • <code>ValidFrom</code> • <code>ValidUntil</code> Las propiedades siguientes de la propiedad <code>SpotFleetRequestConfigData</code> admiten actualizaciones de <code>No interruption</code>: • <code>ExcessCapacityTerminationPolicy</code> • <code>TargetCapacity</code>	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
		AWS::EMR::InstanceGroupConfig CloudFormation es ahora compatible con el escalado automático para grupos de instancias de tareas de Amazon EMR. AWS::Events::Rule La propiedad RoleArn se ha dejado de utilizar en el recurso Rule. Utilice el tipo de propiedad RoleArn en el tipo de propiedad Target para especificar el rol de IAM que hay que utilizar para un destino. AWS::Kinesis::Stream La propiedad ShardCount es ahora compatible con las actualizaciones de No interruption . AWS::Lambda::Function Utilice la propiedad TracingConfig para configurar los ajustes de rastreo para las funciones de Lambda. AWS::Redshift::Cluster, AWS::Redshift::ClusterParameterGroup, AWS::Redshift::ClusterSecurityGroup, y AWS::Redshift::ClusterSubnetGroup Utilice la propiedad Tags para especificar etiquetas de recursos. AWS::RDS::DBCluster Adición del atributo ReadEndpoint.Address a la función intrínseca Fn::GetAtt .	

Cambio	Fecha de lanzamiento	Descripción	Versión de API
		AWS::S3::Bucket Adición del atributo <code>Arn</code> a la función intrínseca <code>Fn::GetAtt</code> .	

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Recursos nuevos	11 de mayo de 2017	Los siguientes recursos nuevos admiten el uso de AWS WAF con un equilibrador de carga de aplicaciones de Elastic Load Balancing (ELB). AWS::WAFRegional::ByteMatchSet Utilice el recurso <code>AWS::WAFRegional::ByteMatchSet</code> para identificar una parte de una solicitud web que se desea inspeccionar. AWS::WAFRegional::IPSet Utilice el recurso <code>AWS::WAFRegional::IPSet</code> para especificar qué solicitudes web permitir o bloquear en función de las direcciones IP de procedencia de las solicitudes. AWS::WAFRegional::Rule Utilice el recurso <code>AWS::WAFRegional::Rule</code> para especificar una combinación de objetos <code>IPSet</code>, <code>ByteMatchSet</code> y <code>SqlInjectionMatchSet</code> que identifican las solicitudes web que se deben permitir, bloquear o contar. AWS::WAFRegional::SizeConstraintSet Utilice el recurso <code>AWS::WAFRegional::SizeConstraintSet</code> para especificar la restricción de tamaño aplicada al comprobar el tamaño de una solicitud web y qué partes de la solicitud hay que comprobar. AWS::WAFRegional::SqlInjectionMatchSet Utilice el recurso <code>AWS::WAFRegional::SqlInjectionMatchSet</code> para permitir, bloquear o contar las solicitudes que contengan	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
		código SQL malicioso en una parte específica de las solicitudes web. AWS::WAFRegional::WebACL Utilice el recurso <code>AWS::WAFRegional::WebACL</code> para identificar las solicitudes web que se desea permitir, bloquear o contar. AWS::WAFRegional::WebACLAssociation Utilice el recurso <code>AWS::WAFRegional::WebACLAssociation</code> para asociar un grupo de control de acceso web (ACL) a un recurso. AWS::WAFRegional::XssMatchSet Utilice el recurso <code>AWS::WAFRegional::XssMatchSet</code> para especificar las partes de solicitudes web que se desea que AWS WAF inspeccione para detectar ataques de scripting entre sitios y el nombre del encabezado que debe inspeccionarse.	

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Recursos nuevos	28 de abril de 2017	AWS::Cognito::IdentityPool Utilice el recurso <code>AWS::Cognito::IdentityPool</code> para crear un grupo de identidades de Amazon Cognito. AWS::Cognito::IdentityPoolRoleAttachment Utilice el recurso <code>AWS::Cognito::IdentityPoolRoleAttachment</code> para administrar la configuración de roles para un grupo de identidades de Amazon Cognito. AWS::Cognito::UserPool Utilice el recurso <code>AWS::Cognito::UserPool</code> para crear un grupo de usuarios de Amazon Cognito. AWS::Cognito::UserPoolClient Utilice el recurso <code>AWS::Cognito::UserPoolClient</code> para crear un cliente de grupo de usuarios. AWS::Cognito::UserPoolGroup Utilice el recurso <code>AWS::Cognito::UserPoolGroup</code> para crear un grupo de usuarios en un grupo de usuarios de Amazon Cognito. AWS::Cognito::UserPoolUser Utilice el recurso <code>AWS::Cognito::UserPoolUser</code> para crear un usuario del grupo de usuarios de Amazon Cognito.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
		<u>AWS::Cognito::UserPoolUserToGroupAttachment</u> Utilice el recurso <code>AWS::Cognito::UserPoolUserToGroupAttachment</code> para asociar un usuario a un grupo de usuarios de Amazon Cognito.	

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Recursos actualizados	28 de abril de 2017	SourceDetails Utilice la subpropiedad <code>MaximumExecutionFrequency</code> del recurso <code>AWS::Config::ConfigRule</code> para ejecutar evaluaciones para una regla personalizada utilizando un disparador periódico. AWS::EC2::Volume Compatibilidad con volúmenes elásticos para Amazon Elastic Block Store (Amazon EBS) en CloudFormation. Compatibilidad con actualizaciones de <code>No interruption</code> en tres propiedades: <code>VolumeType</code>, <code>Size</code> e <code>Iops</code>. AWS::EC2::SecurityGroup Utilice la propiedad <code>GroupName</code> para especificar un nombre para el grupo de seguridad de Amazon EC2. AWS::ECS::Service Tres nuevas propiedades para <code>AWS::ECS::Service</code>: <code>PlacementConstraints</code>, <code>PlacementStrategies</code> y <code>ServiceName</code>. AWS::ECS::TaskDefinition Utilice la propiedad <code>PlacementConstraints</code> para definir las restricciones de ubicación para tareas del servicio. AWS::ElastiCache::ReplicationGroup Adición de los atributos <code>ConfigurationEndpoint.Address</code> y <code>ConfigurationEndpoint.Port</code> a la función intrínseca <code>Fn::GetAtt</code>.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
		AWS::ElasticLoadBalancingV2::LoadBalancer Utilice la propiedad <code>IpAddressType</code> para especificar el tipo de direcciones IP que utilizan las subredes del balanceador de carga. AWS::EMR::Cluster CloudFormation es ahora compatible con el escalado automático para clústeres de Amazon EMR. AWS::IAM::ManagedPolicy Utilice la propiedad <code>ManagedPolicyName</code> para especificar un nombre personalizado para la política administrada de IAM. AWS::Lambda::Function Utilice la propiedad <code>Tags</code> para añadir etiquetas a la función de Lambda. AWS::OpsWorks::Instance Adición de los siguientes atributos a la función intrínseca <code>Fn::GetAtt</code> : <code>AvailabilityZone</code> , <code>PrivateDnsName</code> , <code>PrivateIp</code> y <code>PublicDnsName</code> . AWS::OpsWorks::UserProfile Utilice la propiedad <code>SshUsername</code> para especificar el nombre de SSH de un usuario. Adición del atributo <code>SshUsername</code> a la función intrínseca <code>Fn::GetAtt</code> .	

Cambio	Fecha de lanzamiento	Descripción	Versión de API
		AWS::Redshift::Cluster Utilice la propiedad <code>IamRoles</code> para proporcionar una lista de uno o varios roles de AWS Identity and Access Management que el clúster de Amazon Redshift puede utilizar para acceder a otros servicios de AWS.	
Edición de plantillas en YAML y JSON mediante CloudFormation Designer	6 de abril de 2017	Al crear plantillas de CloudFormation con Designer, ahora puede editar la plantilla en YAML y JSON en el editor integrado. También puede convertir plantillas JSON a YAML y viceversa, dependiendo del lenguaje que prefiera para la creación de la plantilla. Para obtener más información, consulte ¿Qué es CloudFormation Designer? .	2010-05-15
Recurso nuevo	6 de abril de 2017	AWS::SSM::Parameter Utilice el recurso <code>AWS::SSM::Parameter</code> para crear un parámetro de SSM en el almacén de parámetros.	2010-05-15
<code>AWS::Include</code> Transformación	28 de marzo de 2017	Utilice la transformación <code>AWS::Include</code> para hacer referencia a los fragmentos reutilizables almacenados en un bucket de Amazon S3. Para obtener más información, consulte Transformación de AWS::Include .	2010-05-15
Interconexión de Amazon VPC con otra cuenta	28 de marzo de 2017	Ahora puede utilizar CloudFormation para interconectar su Amazon VPC con una VPC de otra cuenta de AWS. Para obtener más información, consulte Peer with an Amazon VPC in Another AWS Account (Interconexión con una Amazon VPC en otra cuenta de).	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Recurso nuevo	28 de marzo de 2017	AWS::ApiGateway::UsagePlanKey Utilice el recurso <code>AWS::ApiGateway::UsagePlanKey</code> para asociar una clave de plan de uso y determinar a qué usuarios se aplica el plan de uso.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Recursos actualizados	28 de marzo de 2017	AWS::EC2::VPCPeeringConnection Utilice la propiedad <code>PeerOwnerId</code> y la propiedad <code>PeerRoleArn</code> para interconectar con una VPC en otra cuenta de AWS. Para obtener más información, consulte Peer with an Amazon VPC in Another AWS Account (Interconexión con una Amazon VPC en otra cuenta de). AWS::IAM::InstanceProfile Utilice la propiedad <code>InstanceProfileName</code> para configurar un perfil de instancia. AWS::Lambda::Function Utilice la propiedad <code>DeadLetterConfig</code> para configurar cómo administra AWS Lambda los eventos que no puede procesar. Node.js v0.10 deja de ser compatible con la propiedad <code>Runtime</code>. AWS::Route53::HealthCheck Hay siete tipos nuevos de subpropiedad de recursos para la propiedad HealthCheckConfig: <code>AlarmIdentifier</code>, <code>ChildHealthChecks</code>, <code>EnableSNI</code>, <code>HealthThreshold</code>, <code>InsufficientDataHealthStatus</code>, <code>Inverted</code> y <code>MeasureLatency</code>. AWS::SQS::Queue Utilice las propiedades <code>ContentBasedDeduplication</code> y <code>FifoQueue</code> para crear colas de	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
		Amazon Simple Queue Service de tipo primero en entrar, primero en salir (FIFO). AWS::S3::Bucket Ahora pueden especificarse nombres de dominio IPv6 para los buckets de Amazon S3.	
Recursos nuevos	10 de febrero de 2017	AWS::StepFunctions::Activity Utilice el recurso <code>AWS::StepFunctions::Activity</code> para crear una actividad de AWS Step Functions. AWS::StepFunctions::StateMachine Utilice el recurso <code>AWS::StepFunctions::StateMachine</code> para crear una máquina de estado de Step Functions.	2010-05-15
Nueva función intrínseca	17 de enero de 2017	Utilice la función <code>Fn::Split</code> para dividir una cadena en una lista de valores de cadena. Para obtener más información, consulte Fn::Split.	2010-05-15
Compatibilidad de la consola con importaciones de listados	17 de enero de 2017	Utilice la consola de CloudFormation para ver todas las pilas que importan un valor de salida exportado. Para obtener más información, consulte Enumeración de pilas que importan un valor de salida exportado.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Recursos actualizados	17 de enero de 2017	AWS::AutoScaling::AutoScalingGroup La propiedad <code>LoadBalancerNames</code> se puede actualizar sin sustituir el grupo de Auto Scaling. AWS::ECS::TaskDefinition Adición de las propiedades <code>NetworkMode</code> y <code>MemoryReservation</code> . AWS::RDS::DBCluster CloudFormation admite actualizaciones de la propiedad <code>Tags</code>. AWS::RDS::DBInstance Adición de la propiedad <code>Timezone</code>. FirehoseAction Adición de la propiedad <code>Separator</code> . AWS::OpsWorks::Instance Adición del atributo <code>PublicIp</code> para la función intrínseca <code>Fn::GetAtt</code> .	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Recursos nuevos	01 de diciembre de 2016	AWS::CodeBuild::Project Utilice el recurso <code>AWS::CodeBuild::Project</code> para crear un proyecto de AWS CodeBuild que defina cómo crea CodeBuild el código fuente. AWS::SSM::Association Utilice el recurso <code>AWS::SSM::Association</code> para asociar un documento de Amazon EC2 Systems Manager a instancias de EC2. AWS::EC2::SubnetCidrBlock Utilice el recurso <code>AWS::EC2::SubnetCidrBlock</code> para asociar un único bloque de CIDR de IPv6 a una subred de Amazon VPC. AWS::EC2::VPCCidrBlock Utilice el recurso <code>AWS::EC2::VPCCidrBlock</code> para asociar un único bloque de CIDR de IPv6 proporcionado por Amazon a una VPC de Amazon VPC.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Recursos actualizados para compatibilidad con IPv6	01 de diciembre de 2016	AWS::EC2::Instance Adición de las propiedades <code>Ipv6AddressCount</code> y <code>Ipv6Addresses</code> . AWS::EC2::NetworkAclEntry Adición de la propiedad <code>Ipv6CidrBlock</code> . AWS::EC2::NetworkInterface Adición de las propiedades <code>Ipv6AddressCount</code> y <code>Ipv6Addresses</code> . AWS::EC2::Route Adición de la propiedad <code>DestinationIpv6CidrBlock</code> . AWS::EC2::SecurityGroupEgress Adición de la propiedad <code>CidrIpv6</code>. AWS::EC2::SecurityGroupIngress Adición de la propiedad <code>CidrIpv6</code>. AWS::EC2::SpotFleet Adición de las propiedades <code>Ipv6AddressCount</code> e <code>Ipv6Addresses</code> para las interfaces de red de la especificación de lanzamiento. AWS::EC2::Subnet Adición del atributo <code>Ipv6CidrBlocks</code> para la función <code>Fn::GetAtt</code> . AWS::EC2::VPC Adición del atributo <code>Ipv6CidrBlocks</code> para la función <code>Fn::GetAtt</code> .	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
		AWS::SSM::Document Adición de la propiedad <code>DocumentType</code> .	
Especificación de recursos	22 de noviembre de 2016	Utilice la especificación de recursos de CloudFormation para crear herramientas que le ayudarán a crear plantillas de CloudFormation. La especificación es un archivo de texto con formato JSON legible electrónicamente. Para obtener más información, consulte Especificación de recursos de CloudFormation .	2010-05-15
Recursos nuevos	22 de noviembre de 2016	AWS::OpsWorks::UserProfile Utilice el recurso <code>AWS::OpsWorks::UserProfile</code> para configurar el acceso de SSH para los usuarios que necesitan obtener acceso a instancias en una pila de OpsWorks. AWS::OpsWorks::Volume Utilice el recurso <code>AWS::OpsWorks::Volume</code> para registrar un volumen de Amazon Elastic Block Store con una pila de OpsWorks.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Recursos actualizados	22 de noviembre de 2016	AWS::OpsWorks::App Adición de la propiedad <code>DataSources</code> . AWS::OpsWorks::Instance Adición de las propiedades <code>BlockDeviceMappings</code> , <code>AgentVersion</code> , <code>ElasticIps</code> , <code>Hostname</code> , <code>Tenancy</code> y <code>Volumes</code>. AWS::OpsWorks::Layer Adición de las propiedades <code>CustomJson</code> y <code>VolumeConfigurations</code> . AWS::OpsWorks::Stack Adición de las propiedades <code>ElasticIps</code> , <code>EcsClusterArn</code> , <code>RdsDbInstances</code> , <code>CloneAppIds</code> , <code>ClonePermissions</code> y <code>SourceStackId</code> . AWS::RDS::DBInstance Adición de la propiedad <code>CopyTagsToSnapshot</code> .	2010-05-15
Importaciones de listas	22 de noviembre de 2016	Importaciones de listas de un valor de salida exportado para realizar un seguimiento de las pilas de CloudFormation que importan el valor. Para obtener más información, consulte Enumeración de pilas que importan un valor de salida exportado.	2010-05-15
Transformaciones	17 de noviembre de 2016	Especifique el AWS Serverless Application Model (AWS SAM) que utiliza CloudFormation para procesar la sintaxis de AWS SAM para aplicaciones sin servidor. Para obtener más información, consulte Transformación.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Recurso nuevo	17 de noviembre de 2016	AWS::SNS::Subscription Utilice el recurso <code>AWS::SNS::Subscription</code> para suscribir un punto de enlace a un tema de Amazon Simple Notification Service.	2010-05-15
Recurso actualizado	17 de noviembre de 2016	AWS::Lambda::Function Utilice la propiedad <code>Environment</code> para especificar pares clave-valor (variables de entorno) a las que una función de AWS Lambda puede obtener acceso. Utilice la propiedad <code>KmsKeyArn</code> para especificar una clave de KMS que AWS Lambda utiliza para cifrar y descifrar variables de entorno.	2010-05-15
Nuevos comandos de la CLI	17 de noviembre de 2016	Carga de artefactos locales en un bucket de S3 Utilice el comando <code>package</code> para cargar artefactos locales a los que se hace referencia en una plantilla de CloudFormation a un bucket de S3. Implementación rápida de plantillas con transformaciones Utilice el comando <code>deploy</code> para combinar las acciones de creación y ejecución de un conjunto de cambios en un solo comando. Este comando es útil para crear o actualizar rápidamente las pilas que contienen transformaciones.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Recurso actualizado	03 de noviembre de 2016	AWS::CloudFront::Distribution Para la propiedad DistributionConfig, utilice la propiedad <code>HttpVersion</code> para especificar la versión HTTP más reciente que los usuarios pueden utilizar para comunicarse con Amazon CloudFront. Para la propiedad ForwardedValues, utilice la propiedad <code>QueryStringCacheKeys</code> para especificar los parámetros de la cadena de consultas que utiliza CloudFront para determinar qué contenido almacenar en la memoria caché.	2010-05-15
Exportaciones de pilas de listas	03 de noviembre de 2016	Utilice la consola de CloudFormation, la API o la AWS CLI para ver una lista de todos los valores de salida exportados para una región. Para obtener más información, consulte Exportación de valores de salida de pila .	2010-05-15
Entrega continua con pilas	03 de noviembre de 2016	Utilice AWS CodePipeline para crear flujos de trabajo de entrega continua con pilas de CloudFormation. Para obtener más información, consulte Entrega continua con CodePipeline .	2010-05-15
Omisión de recursos durante la restauración	03 de noviembre de 2016	Cuando una pila se encuentra en el estado <code>UPDATE_ROLLBACK_FAILED</code> , utilice el parámetro <code>ResourcesToSkip</code> para la acción <code>ContinueUpdateRollback</code> para omitir los recursos que CloudFormation no puede restaurar. Para obtener más información, consulte la sección de solución de problemas en Error de restauración de actualización .	2010-05-15
Mejora de conjuntos de cambios	03 de noviembre de 2016	Puede crear una nueva pila mediante un conjunto de cambios .	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Recurso actualizado	12 de octubre de 2016	AWS::ElastiCache::CacheCluster Actualización de la propiedad <code>CacheNodeType</code> sin sustituir el clúster. AWS::ElastiCache::ReplicationGroup Puede crear un grupo de replicación Redis (modo clúster habilitado) que puede contener varios grupos de nodo (fragmentos), cada uno con un clúster principal y réplicas de lectura. AWS::ElastiCache::SubnetGroup Utilice la propiedad <code>CacheSubnetGroupName</code> para especificar un nombre para un grupo de subredes de Amazon ElastiCache.	2010-05-15
Recursos nuevos	06 de octubre de 2016	AWS::ApiGateway::UsagePlan Utilice el recurso <code>AWS::ApiGateway::UsagePlan</code> para especificar un plan de uso para las API de Amazon API Gateway implementadas. AWS::CodeCommit::Repository Utilice el recurso <code>AWS::CodeCommit::Repository</code> para crear un repositorio de CodeCommit alojado por Amazon Web Services.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Recursos actualizados	06 de octubre de 2016	AWS::ApiGateway::Authorizer Utilice la propiedad <code>ProviderARNs</code> para utilizar grupos de usuario de Amazon Cognito como autorizadores de API de Amazon API Gateway. AWS::ApiGateway::Deployment La propiedad <code>StageName</code> ya no es necesaria. AWS::ElasticLoadBalancingV2::TargetGroup Para la función <code>GetAtt</code>, utilice el atributo <code>LoadBalancerArns</code> para recuperar los nombres de recursos de Amazon (ARN) de los balanceadores de carga que dirigen el tráfico al grupo de destino. AWS::RDS::DBInstance Utilice las propiedades <code>Domain</code> y <code>DomainIAMRoleName</code> para utilizar la autenticación de Windows cuando los usuarios se conectan a la instancia de base de datos de RDS. AWS::EC2::SecurityGroupEgress Utilice la propiedad <code>DestinationPrefixListId</code> para especificar el prefijo del servicio de AWS de un punto de conexión de Amazon VPC.	2010-05-15
Mejora de la referencia cruzada de pilas	06 de octubre de 2016	Utilice las funciones intrínsecas para personalizar el valor <code>Name</code> de una exportación o para hacer referencia a un valor en la función ImportValue .	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Rol de servicio de CloudFormation	26 de septiembre de 2016	Uso de un rol de servicio de AWS Identity and Access Management (IAM) para las operaciones de pila de CloudFormation. CloudFormation utiliza las credenciales del rol para realizar llamadas a los recursos de la pila en su nombre. Para obtener más información, consulte Rol de servicio de CloudFormation .	2010-05-15
Nueva característica	19 de septiembre de 2016	Puede utilizar el campo de salida Export y la función intrínseca Fn::ImportValue para que una pila consulte las salidas de recursos en otra pila. Para obtener más información, consulte Salidas , Fn::ImportValue y Tutorial: Consulte las salidas de recursos en otra pila de CloudFormation .	2010-05-15
Compatibilidad con YAML	19 de septiembre de 2016	Puede utilizar el formato YAML para crear plantillas de CloudFormation. YAML también le permite, por ejemplo, añadir comentarios a las plantillas o utilizar la forma abreviada para funciones intrínsecas. Para obtener más información, consulte Formato de plantilla de CloudFormation .	2010-05-15
Nueva función intrínseca	19 de septiembre de 2016	Utilice la función Fn::Sub para sustituir variables en una cadena de entrada por los valores que especifique. Para obtener más información, consulte Fn::Sub .	2010-05-15
Recursos nuevos	19 de septiembre de 2016	AWS::KMS::Alias Utilice el recurso de AWS::KMS::Alias para crear un alias para una AWS KMS key.	

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Recursos actualizados	19 de septiembre de 2016	AWS::EC2::SpotFleet Para la propiedad <code>LaunchSpecifications</code> , utilice la propiedad <code>SpotPrice</code> para especificar el precio de la puja para cada tipo de instancia. AWS::ECS::Cluster Utilice la propiedad <code>ClusterName</code> para especificar un nombre para un clúster de Amazon Elastic Container Service. AWS::ECS::TaskDefinition Utilice la propiedad <code>TaskRoleArn</code> para especificar un rol de AWS Identity and Access Management que los contenedores de Amazon Elastic Container Service utilizan para hacer llamadas de AWS en su nombre. Utilice la propiedad <code>Family</code> para registrar una definición de tarea para una familia específica. AWS::Elasticsearch::Domain Utilice la propiedad <code>ElasticsearchVersion</code> para especificar qué versión de Elasticsearch utilizar.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Recursos nuevos	11 de agosto de 2016	Use los siguientes recursos de Balanceador de carga de aplicaciones de Elastic Load Balancing para distribuir el tráfico de aplicaciones entrantes a varios destinos, por ejemplo, instancias de EC2, en varias zonas de disponibilidad: • AWS::ElasticLoadBalancingV2::Listener • AWS::ElasticLoadBalancingV2::ListenerRule • AWS::ElasticLoadBalancingV2::LoadBalancer • AWS::ElasticLoadBalancingV2::TargetGroup	2010-05-15
Recurso actualizado	11 de agosto de 2016	AWS::AutoScaling::AutoScalingGroup Utilice la propiedad <code>TargetGroupARNs</code> para asociar el grupo de Auto Scaling con uno o varios grupos de destino de Balanceador de carga de aplicaciones. AWS::ECS::Service Para la propiedad <code>LoadBalancers</code> de carga, utilice la propiedad <code>TargetGroupArn</code> para asociar un servicio de Amazon Elastic Container Service con un grupo de destino de Balanceador de carga de aplicaciones.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Recursos nuevos	09 de agosto de 2016	CloudFormation añadió los siguientes recursos: AWS::ApplicationAutoScaling::ScalableTarget y AWS::ApplicationAutoScaling::ScalingPolicy Uso de una política de escalado de Application Auto Scaling para definir cuándo y cómo se adapta un recurso de destino. AWS::CertificateManager::Certificate Aprovisiona un certificado de AWS Certificate Manager que puede utilizar con otros servicios de AWS para permitir conexiones seguras.	2010-05-15
Recursos actualizados	09 de agosto de 2016	CloudFormation actualizó los siguientes recursos: AWS::CloudFront::Distribution Para la propiedad <code>ViewerCertificate</code> de configuración de la distribución, puede especificar un certificado de AWS Certificate Manager. Para la propiedad <code>Origin</code> de configuración de la distribución, puede especificar encabezados personalizados y los protocolos SSL para orígenes personalizados. AWS::EFS::FileSystem Puede especificar el modo de desempeño para un sistema de archivos de Amazon Elastic File System.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Recursos nuevos	20 de julio de 2016	AWS IoT Use AWS IoT para declarar una política de AWS IoT, un certificado X.509, una asociación entre una política y un principal (un certificado X.509 u otra credencial), un objeto de AWS IoT, una asociación entre un principal y un objeto o una regla de AWS IoT. • AWS::IoT::Certificate • AWS::IoT::Policy • AWS::IoT::PolicyPrincipalAttachment • AWS::IoT::Thing • AWS::IoT::ThingPrincipalAttachment • AWS::IoT::TopicRule	2010-05-15
Recursos actualizados	20 de julio de 2016	CloudFormation actualizó los siguientes recursos: AWS::IAM::Group, AWS::IAM::Role, AWS::IAM::User Utilice las propiedades del nombre para especificar un nombre personalizado para los recursos de AWS Identity and Access Management (IAM). AWS::ApiGateway::Method Para la propiedad <code>Integration</code> puede utilizar la propiedad <code>PassthroughBehavior</code> para especificar cuándo Amazon API Gateway transfiere solicitudes dirigidas al backend de destino. AWS::ApiGateway::Model y AWS::ApiGateway::RestApi Puede especificar objetos JSON para las propiedades <code>Schema</code> y <code>Body</code>.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
UpdatePolicy para grupo de Auto Scaling	9 de junio de 2016	Para el atributo UpdatePolicy , utilice la propiedad AutoScalingReplacingUpdate para especificar si un grupo de Auto Scaling y las instancias que contiene se sustituirán cuando actualice el grupo de Auto Scaling. Durante una sustitución, CloudFormation conserva el antiguo grupo de escalado automático hasta que crea el nuevo de forma satisfactoria para AWS CloudFormation pueda volver al antiguo grupo de escalado automático si falla la actualización. Para obtener más información, consulte Atributo UpdatePolicy .	2010-05-15
Recurso nuevo	9 de junio de 2016	CloudFormation añadió los siguientes recursos: AWS::EC2::FlowLog Crea un registro de flujo de Amazon Elastic Compute Cloud que captura el tráfico IP para una interfaz de red, la subred o VPC especificadas. AWS::KinesisFirehose::DeliveryStream Crea una secuencia de entrega que proporciona datos en streaming en tiempo real a un destino, como Amazon Simple Storage Service, Amazon Redshift o Amazon OpenSearch Service.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Recursos actualizados	9 de junio de 2016	CloudFormation actualizó los siguientes recursos: AWS::Kinesis::Stream Utilice la propiedad Name para especificar un nombre para una secuencia de Amazon Kinesis. AWS::Lambda::Function Para la propiedad Code, puede utilizar la propiedad ZipFile y el módulo de respuesta cfn para entornos de tiempo de ejecución nodejs4.3 . AWS::SNS::Topic CloudFormation habilita actualizaciones para el recurso del tema de Amazon Simple Notification Service.	2010-05-15
Recurso nuevo	25 de abril de 2016	Utilice el recurso AWS::EC2::Host para asignar un servidor físico totalmente dedicado para el lanzamiento de instancias de EC2.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Recursos actualizados	25 de abril de 2016	AWS::EC2::Instance Utilice las propiedades <code>Affinity</code> y <code>HostId</code> para lanzar instancias en un host dedicado de Amazon Elastic Compute Cloud. AWS::ECS::Service Utilice la propiedad <code>DeploymentConfiguration</code> para configurar cuántas tareas pueden ejecutarse durante una implementación. AWS::ECS::TaskDefinition CloudFormation añadió compatibilidad para propiedades adicionales de la definición de contenedor de Amazon Elastic Container Service. AWS::GameLift::Fleet Utilice las propiedades <code>MaxSize</code> y <code>MinSize</code> para especificar la cantidad mínima y máxima de instancias de EC2 permitidas en la flota de Amazon GameLift Servers. AWS::Lambda::Function Utilice la propiedad <code>FunctionName</code> para especificar un nombre para su función AWS Lambda. También puede utilizar Python 2.7 para especificar una función en línea.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Recursos nuevos	18 de abril de 2016	Amazon API Gateway Uso de los recursos de Amazon API Gateway para publicar, mantener y monitorizar los API a cualquier escala. Puede crear los API que los clientes pueden llamar para obtener acceso a sus servicios de back-end, como aplicaciones que ejecutan instancias de EC2 o código ejecutado en 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 Cree una regla Eventos de Amazon CloudWatch que supervise los cambios en los recursos de AWS en su cuenta (eventos). Si un evento entrante coincide con las condiciones que describió en la regla, Eventos de Amazon CloudWatch envía mensajes a sus destinos específicos, como funciones de AWS Lambda o temas de Amazon Simple Notification Service.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
		AWS::WAF::SizeConstraintSet y AWS::WAF::XssMatchSet Utilice las dos reglas de AWS WAF para comprobar el tamaño de una solicitud web o para evitar los ataques de scripting entre sitios.	
Recursos nuevos	31 de marzo de 2016	Utilice el recurso AWS::Lambda::Alias para crear alias para sus funciones AWS Lambda y el recurso AWS::Lambda::Version para crear versiones de sus funciones.	2010-05-15
Recursos actualizados	31 de marzo de 2016	CloudFormation actualizó los siguientes recursos: AWS::EMR::Cluster y AWS::EMR::InstanceGroupConfig Utilice la propiedad <code>EbsConfiguration</code> para configurar volúmenes de almacenamiento de Amazon Elastic Block Store para sus grupos de instancias o clústeres de Amazon EMR. AWS::Lambda::Function Utilice la propiedad <code>VpcConfig</code> para permitir que las funciones AWS Lambda obtengan acceso a recursos de una VPC. AWS::S3::Bucket Para las reglas de ciclo de vida de Amazon Simple Storage Service, puede especificar varias reglas de transición que especifican cuándo realizan los objetos la transición a una clase de almacenamiento especificado.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Conjuntos de cambios	29 de marzo de 2016	Antes de actualizar pilas, uso de los conjuntos de cambios para ver cómo podrían afectar los cambios a los recursos que ya se están ejecutando. Para obtener más información, consulte Actualización de pilas con conjuntos de cambios .	2010-05-15
Recursos nuevos	15 de marzo de 2016	Use los recursos AWS::GameLift::Alias , AWS::GameLift::Build y AWS::GameLift::Fleet para implementar servidores de juegos multijugador en AWS.	2010-05-15
Recursos nuevos	26 de febrero de 2016	CloudFormation añadió los siguientes recursos: AWS::ECR::Repository Cree repositorios de Amazon Elastic Container Registry donde los usuarios puedan insertar y extraer imágenes de Docker. AWS::EC2::NatGateway Utilice el gateway de conversión de direcciones de red (NAT) para habilitar las instancias de EC2 en una subred privada para conectarse a Internet. AWS::Elasticsearch::Domain Cree dominios de Amazon OpenSearch Service que ejecuten clústeres de OSS de Elasticsearch heredados. AWS::EMR::Cluster, AWS::EMR::InstanceGroupConfig, AWS::EMR::Step Use los recursos de Amazon EMR para ayudarle a analizar y procesar grandes cantidades de datos. Puede crear clústeres y, a continuación, ejecutar trabajos en ellos.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Recursos actualizados	26 de febrero de 2016	CloudFormation actualizó los siguientes recursos: AWS::CloudTrail::Trail Utilice la propiedad <code>IsMultiRegionTrail</code> para especificar si desea crear un registro de seguimiento de AWS CloudTrail en la región en la que ha creado una pila o en todas las regiones. AWS::Config::ConfigurationRecorder En el grupo de registro, utilice la propiedad <code>IncludeGlobalResourceTypes</code> para registrar todos los tipos de recursos globales. AWS::RDS::DBCluster Utilice las propiedades <code>KmsKeyId</code> y <code>StorageEncrypted</code> para cifrar instancias de la base de datos en el clúster.	2010-05-15
Mantenga recursos	26 de febrero de 2016	Para pilas en el estado <code>DELETE_FAILED</code>, utilice el parámetro <code>RetainResources</code> para conservar los recursos que CloudFormation no puede eliminar. Para obtener más información, consulte Error al eliminar la pila.	2010-05-15
Actualización de las etiquetas de pilas	26 de febrero de 2016	Puede añadir, modificar o eliminar etiquetas de pilas al actualizar una pila. Para obtener más información, consulte Actualizaciones de pilas de CloudFormation.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Continuación de anulaciones de restauraciones de actualizaciones fallidas	25 de enero de 2016	Para una pila en el estado UPDATE_ROLLBACK_FAILED , puede continuar anulando la actualización para que la pila vuelva a estar operativa. De esta forma, puede devolver la pila a su configuración original y tratar de volver a actualizarla. Para obtener más información, consulte Continuación de la restauración de una actualización .	2010-05-15
Nuevas plantillas de ejemplo disponibles para la región Asia Pacífico (Seúl).	7 de enero de 2016	La siguiente colección de plantillas de muestra de CloudFormation son para la región ap-northeast-2: • Soluciones de ejemplo • Marcos de aplicaciones • Services Para obtener más información, consulte Trabajar con plantillas de CloudFormation.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Recursos nuevos	28 de diciembre de 2015	CloudFormation añadió los siguientes recursos: AWS::DirectoryService::MicrosoftAD Utilice el recurso Microsoft Active Directory para crear un directorio de Microsoft Active Directory en AWS. AWS::Logs::Destination y AWS::Logs::LogStream Uso de los recursos de Amazon CloudWatch Logs para crear un destino para el procesamiento en tiempo real de datos de registro o para crear flujos de registro, respectivamente. AWS::WAF::ByteMatchSet, AWS::WAF::IPSet, AWS::WAF::Rule, AWS::WAF::SqlInjectionMatchSet, y AWS::WAF::WebACL Uso de los recursos de AWS WAF para controlar y monitorizar solicitudes web para su contenido.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Actualizaciones de recursos	28 de diciembre de 2015	CloudFormation actualizó los siguientes recursos: AWS::CloudFront::Distribution Para la configuración de la distribución, utilice la propiedad <code>WebACLIId</code> para asociar una lista de control de acceso (ACL) web de AWS WAF con una distribución de Amazon CloudFront. Para el comportamiento de la caché y el comportamiento predeterminado de la caché, puede especificar un valor máximo y un valor predeterminado de periodo de tiempo de vida (TTL). AWS::DynamoDB::Table Puede crear, actualizar o eliminar un índice secundario o global sin sustituir su tabla de Amazon DynamoDB. AWS::S3::Bucket Utilice la propiedad <code>ReplicationConfiguration</code> para especificar qué objetos replicar y dónde se almacenan. Utilice las propiedades en la propiedad <code>NotificationConfiguration</code> para especificar filtros para que Amazon Simple Storage Service envíe notificaciones para los objetos que especifique.	2010-05-15
Agrupamiento y clasificación de parámetros	3 de diciembre de 2015	Utilice la clave de metadatos AWS::CloudFormation::Interface para agrupar y clasificar parámetros en la consola de CloudFormation cuando los usuarios crean o actualizan una pila con su plantilla.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Atributo de política de actualización	3 de diciembre de 2015	Para un atributo de política de actualización de Auto Scaling, utilice la propiedad <code>MinSuccessfulInstancesPercent</code> para especificar el porcentaje de instancias que deben enviar una señal de éxito para una actualización correcta.	2010-05-15
Recursos nuevos	3 de diciembre de 2015	CloudFormation añadió los siguientes recursos: AWS::CodePipeline::Pipeline y AWS::CodePipeline::CustomActionType Uso de los recursos de CodePipeline para crear una canalización que describa la forma en la que se someten los cambios a un proceso de lanzamiento. AWS::Config::ConfigurationRecorder, AWS::Config::DeliveryChannel, y AWS::Config::ConfigRule Use los recursos de AWS Config para supervisar los cambios de configuración en determinados recursos de AWS. AWS::KMS::Key Utilice el recurso AWS Key Management Service (AWS KMS) para crear las claves administradas por el cliente en AWS KMS que los usuarios pueden utilizar para cifrar pequeñas cantidades de datos. AWS::SSM::Document Uso de Amazon EC2 Systems Manager para crear un documento que especifica las configuraciones en la instancia.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Actualización de recursos	3 de diciembre de 2015	CloudFormation actualizó los siguientes recursos: AWS::AutoScaling::LaunchConfiguration Especifique si los volúmenes de EBS están cifrados. AWS::AutoScaling::ScalingPolicy Puede utilizar dos tipos diferentes de políticas (sencillo y escalado de pasos) para especificar cómo se adapta un grupo de Auto Scaling cuando se infringe una alarma de Amazon CloudWatch (CloudWatch). AWS::CloudTrail::Trail Utilice las propiedades de CloudWatch para enviar registros a un grupo de registros de CloudWatch. Puede agregar etiquetas a un registro de seguimiento y especificar la AWS KMS key que desea usar para cifrar los registros. AWS::CodeDeploy::Application, AWS::CodeDeploy::DeploymentConfig, y AWS::CodeDeploy::DeploymentGroup Utilice las propiedades <code>ApplicationName</code> , <code>DeploymentConfigName</code> y <code>DeploymentGroupName</code> para especificar nombres personalizados para recursos de CodeDeploy. AWS::DynamoDB::Table Utilice la propiedad <code>StreamSpecification</code> para especificar la configuración para capturar cambios en los elementos almacenados en una tabla de Amazon DynamoDB (DynamoDB).	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
		AWS::EC2::Instance Utilice la propiedad <code>SsmAssociations</code> para asociar un documento de Amazon EC2 Systems Manager con una instancia. AWS::EC2::SpotFleet Utilice la propiedad <code>AllocationStrategy</code> para especificar cómo asignar la capacidad de destino en los grupos de spot. Utilice la propiedad <code>ExcessCapacityTerminationPolicy</code> para especificar el modo en que se terminan las instancias si la capacidad de destino se encuentra por debajo del tamaño de la flota de spot. AWS::Redshift::Cluster Utilice la propiedad <code>KmsKeyId</code> para especificar una AWS KMS key para cifrar los datos en un clúster de Amazon Redshift. AWS::WorkSpaces::Workspace Utilice las propiedades de cifrado para cifrar datos almacenados en volúmenes.	
Actualización de recursos	4 de noviembre de 2015	Para el recurso AWS::EC2::Volume , utilice la propiedad <code>AutoEnableIO</code> para reanudar automáticamente las operaciones de E/S si un volumen de datos se vuelve incoherente.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Recursos nuevos	1 de octubre de 2015	CloudFormation añadió los siguientes recursos: AWS::CodeDeploy::Application, AWS::CodeDeploy::DeploymentGroup, y AWS::CodeDeploy::DeploymentConfig Uso de los recursos de CodeDeploy para crear y aplicar las implementaciones en instancias de EC2 o en las instalaciones. AWS::DirectoryService::SimpleAD Utilice el recurso Simple Active Directory para crear un AD sencillo de Directory Service, que es un directorio compatible con Microsoft Active Directory. AWS::EC2::PlacementGroup Uso de un grupo de ubicación para crear un clúster de instancias en una red de baja latencia. AWS::EC2::SpotFleet Uso de una flota de spot para lanzar una colección de instancias de spot que ejecutan tareas que se pueden interrumpir. AWS::Lambda::EventSourceMapping Utilice el recurso de asignación de origen de eventos para especificar una secuencia como un origen de evento para una función de AWS Lambda (Lambda). AWS::Lambda::Permission Use un permiso de Lambda para añadir una declaración a la política de una función de Lambda.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
		AWS::Logs::SubscriptionFilter Utilice el filtro de suscripción para definir los eventos de registros que se entregan a su flujo de Kinesis. AWS::RDS::DBCluster y AWS::RDS::DBClusterParameterGroup Utilice el clúster y los recursos del grupo de parámetros del clúster para crear un clúster de base de datos de Amazon Aurora. AWS::WorkSpaces::Workspace Uso de WorkSpaces para crear experiencias de escritorio basadas en la nube.	

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Actualizaciones de recursos	1 de octubre de 2015	CloudFormation actualizó los siguientes recursos: AWS::ElastiCache::ReplicationGroup Utilice la función intrínseca <code>Fn::GetAtt</code> para obtener una lista de puertos y direcciones de réplica de solo lectura. AWS::OpsWorks::Stack Utilice la propiedad <code>AgentVersion</code> para especificar un agente de OpsWorks determinado. AWS::OpsWorks::App Utilice la propiedad <code>Environment</code> para especificar variables del entorno para una aplicación de OpsWorks. AWS::S3::Bucket Para la propiedad NotificationConfiguration, puede configurar los ajustes de las notificaciones para funciones de Lambda y colas de Amazon Simple Queue Service (Amazon SQS).	2010-05-15
Claves de condición de IAM	1 de octubre de 2015	Para las políticas de AWS Identity and Access Management (IAM), use claves de condición específicas de CloudFormation para especificar cuando entra en vigor una política de IAM. Para obtener más información, consulte Controlar el acceso con AWS Identity and Access Management.	2010-05-15
CloudFormation Designer	1 de octubre de 2015	Use CloudFormation Designer para crear y modificar las plantillas mediante una interfaz de arrastrar y colocar.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Recurso nuevo	24 de agosto de 2015	Use el recurso AWS::EC2::VPCEndpoint para establecer una conexión privada entre su VPC y otro servicio de AWS.	2010-05-15
Actualizaciones de recursos	24 de agosto de 2015	CloudFormation actualizó los siguientes recursos: AWS::ElasticBeanstalk::Environment Utilice la propiedad Tags para especificar etiquetas (pares clave-valor) para un entorno de AWS Elastic Beanstalk (Elastic Beanstalk). AWS::Lambda::Function Para la propiedad Code, utilice la propiedad ZipFile para escribir el código fuente de la función de Lambda directamente en una plantilla. En la actualidad, puede utilizar la propiedad ZipFile solo para entornos de tiempo de ejecución node.js. Todavía puede apuntar a un archivo en un bucket de S3 para todos los entornos de tiempo de ejecución, por ejemplo, java8 y node.js. AWS::OpsWorks::Instance Utilice la propiedad EbsOptimized para indicar si se ha optimizado una instancia para E/S de Amazon Elastic Block Store (Amazon EBS). AWS::RDS::DBInstance Para la propiedad SourceDBInstanceId, puede especificar una instancia de base de datos en otra región para crear una réplica de lectura entre regiones.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
URL de plantilla de Amazon S3	24 de agosto de 2015	En el caso de buckets habilitados para el control de versiones, puede especificar un ID de versión en una URL de plantilla de Amazon S3 al crear o actualizar una pila, por ejemplo <code>https://s3.amazonaws.com/templates/myTemplate.template?versionId=123ab1cdeKd0W5IH4GAcYbEngcpTJTdW</code> .	2010-05-15
Recurso nuevo	3 de agosto de 2015	Use el recurso AWS::EFS::FileSystem para crear un sistema de archivos de Amazon Elastic File System (Amazon EFS) y el recurso AWS::EFS::MountTarget para crear un punto de montaje para un sistema de archivos.	2010-05-15
Cambio en la necesidad de permisos	11 de junio de 2015	Al crear o actualizar un recurso AWS::RDS::DBInstance , ahora también tiene que tener permiso para llamar a la acción <code>ec2:DescribeAccountAttributes</code> .	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Recursos nuevos	11 de junio de 2015	CloudFormation añadió los siguientes recursos: AWS::DataPipeline::Pipeline Uso de canalizaciones de datos para automatizar el movimiento y la transformación de los datos. Recursos Amazon Elastic Container Service Uso de los recursos AWS::ECS::Service, AWS::ECS::Cluster y AWS::ECS::TaskDefinition para crear contenedores de Docker en un clúster de instancias de EC2. AWS::ElastiCache::ReplicationGroup Uso de grupos de replicación para crear una colección de nodos con un clúster de lectura-escritura principal y un máximo de cinco clústeres de solo lectura secundario. AWS::IAM::ManagedPolicy Use políticas administradas para crear políticas en su cuenta de AWS que puede usar para aplicar permisos para usuarios, grupos y roles de IAM. AWS::Lambda::Function Uso de funciones de Lambda para ejecutar el código en respuesta a eventos. AWS::RDS::OptionGroup Use grupos de opciones como ayuda para crear y administrar bases de datos de Amazon Relational Database Service (Amazon RDS).	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Actualizaciones de recursos	11 de junio de 2015	CloudFormation actualizó los siguientes recursos: AWS::EC2::Subnet Utilice la propiedad <code>MapPublicIpOnLaunch</code> para asignar de forma automática direcciones IP públicas a instancias de una subred. AWS::ElastiCache::CacheCluster Utilice la propiedad <code>SnapshotName</code> para restaurar datos de snapshot en un nuevo clúster de caché de Redis. AWS::IAM::User Para la propiedad <code>LoginProfile</code>, utilice la propiedad <code>PasswordResetRequired</code>, a fin de que los usuarios deban establecer una nueva contraseña cuando inicien sesión en la Consola de administración de AWS. AWS::OpsWorks::Layer Utilice la propiedad <code>LifecycleEventConfiguration</code> para configurar los eventos del ciclo de vida para una capa de OpsWorks. AWS::S3::Bucket Para la propiedad <code>LifecycleConfiguration</code>, utilice las propiedades <code>NoncurrentVersionExpirationInDays</code> y <code>NoncurrentVersionTransition</code> para especificar reglas de ciclo de vida para las versiones de objetos no actuales.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Nuevos tipos de parámetros	19 de mayo de 2015	Cuando utilice la consola de CloudFormation para crear o actualizar una pila, puede buscar valores por tipos de parámetros específicos de AWS por ID, el nombre o valor de la etiqueta del nombre. CloudFormation también añadió compatibilidad para los siguientes tipos de parámetros específicos de AWS. Para obtener más información, consulte Parámetros. • <code>AWS::EC2::AvailabilityZone::Name</code> • <code>List<AWS::EC2::AvailabilityZone::Name></code> • <code>AWS::EC2::Instance::Id</code> • <code>List<AWS::EC2::Instance::Id></code> • <code>AWS::EC2::Image::Id</code> • <code>List<AWS::EC2::Image::Id></code> • <code>AWS::EC2::SecurityGroup::GroupName</code> • <code>List<AWS::EC2::SecurityGroup::GroupName></code> • <code>AWS::EC2::Volume::Id</code> • <code>List<AWS::EC2::Volume::Id></code> • <code>AWS::Route53::HostedZone::Id</code> • <code>List<AWS::Route53::HostedZone::Id></code>	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Recursos nuevos	16 de abril de 2015	CloudFormation añadió los siguientes recursos: AWS::AutoScaling::LifecycleHook Uso de enlaces del ciclo de vida de Auto Scaling para controlar el estado de una instancia después de lanzarla o terminarla. AWS::RDS::EventSubscription Uso de suscripciones a eventos para obtener notificaciones sobre eventos de Amazon RDS.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Actualizaciones de recursos	16 de abril de 2015	CloudFormation actualizó los siguientes recursos: AWS::AutoScaling::AutoScalingGroup Utilice la propiedad <code>NotificationConfigurations</code> para especificar varias notificaciones. AWS::AutoScaling::LaunchConfiguration Utilice la propiedad <code>PlacementTenancy</code> para especificar la tenencia de instancias. Utilice las propiedades <code>ClassicLinkVPCId</code> y <code>ClassicLinkVPCSecurityGroups</code> para enlazar instancias EC2-Classic a una VPC habilitada para ClassicLink. AWS::AutoScaling::ScalingPolicy Utilice la propiedad <code>MinAdjustmentStep</code> para especificar el número mínimo de instancias que se añaden o eliminan durante un evento de escalado. AWS::CloudFront::Distribution Para los certificados del usuario, utilice la propiedad <code>MinimumProtocolVersion</code> para especificar una versión mínima de protocolo. Para comportamientos de la caché, utilice la propiedad <code>CachedMethods</code> para especificar qué métodos Amazon CloudFront (CloudFront) almacena en caché las respuestas. Para orígenes, use <code>OriginPath</code> para especificar una ruta que CloudFront utiliza para solicitar contenido.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
		<u>AWS::ElastiCache::CacheCluster</u> Para clústeres de caché de Memcached, utilice las propiedades <code>AZMode</code> y <code>PreferredAvailabilityZones</code> para especificar nodos en varias zonas de disponibilidad (AZ). <u>AWS::EC2::Volume</u> Utilice la propiedad <code>KmsKeyId</code> para especificar una clave administrada para los volúmenes cifrados. <u>AWS::OpsWorks::Instance</u> Utilice la propiedad <code>TimeBasedAutoScaling</code> para escalar automáticamente instancias según el calendario que usted especifique. <u>AWS::OpsWorks::Layer</u> Utilice la propiedad <code>LoadBasedAutoScaling</code> para especificar políticas de escalado basadas en la carga. Para las configuraciones de volumen, utilice las propiedades <code>VolumeType</code> e <code>Iops</code> para especificar un tipo de volumen y el número de operaciones de E/S por segundo, respectivamente. <u>AWS::RDS::DBInstance</u> Utilice la propiedad <code>CharacterSetName</code> para especificar un conjunto de caracteres para los motores de bases de datos admitidos. Utilice la propiedad <code>StorageEncrypted</code> para indicar si las instancias de la base de datos se cifrarán y el <code>KmsKeyId</code> para especificar una clave administrada por el cliente para instancias de bases de datos cifradas.	

Cambio	Fecha de lanzamiento	Descripción	Versión de API
		AWS::Route53::HealthCheck Utilice la propiedad <code>HealthCheckTags</code> para asociar etiquetas con comprobaciones de estado. AWS::Route53::HostedZone Utilice la propiedad <code>VPCs</code> para crear zonas hospedadas privadas. Utilice la propiedad <code>HostedZoneTags</code> para asociar etiquetas con zonas hospedadas.	
Nueva sección de plantillas	16 de abril de 2015	Añada la sección Metadatos a sus plantillas para incluir los objetos JSON arbitrarios que describen sus plantillas, por ejemplo, el diseño o los detalles de implementación.	2010-05-15
Actualización de recursos	8 de abril de 2015	Para el recurso AWS::CloudFormation::CustomResource , puede especificar Nombres de recurso de Amazon (ARN) de funciones de Lambda en la propiedad <code>ServiceToken</code> .	2010-05-15
Amazon RDS update	24 de diciembre de 2014	CloudFormation añadió dos nuevas propiedades para instancias de base de datos de RDS. Puede asociar un grupo de opciones con una instancia de base de datos y especificar el tipo de almacenamiento de la instancia de la base de datos. Para obtener más información, consulte AWS::RDS::DBInstance .	2010-05-15
Elastic Load Balancing update	24 de diciembre de 2014	Puede utilizar la propiedad <code>ConnectionSettings</code> para especificar durante cuánto tiempo pueden permanecer inactivas las conexiones. Para obtener más información, consulte AWS::ElasticLoadBalancing::LoadBalancer .	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Actualización de Route 53	6 de noviembre de 2014	Ahora puede aprovisionar y administrar zonas hospedadas , comprobaciones de estado , conjuntos de registros de conmutación por error y conjuntos de registros de geolocalización de Route 53.	2010-05-15
Mejora de la actualización continua de Auto Scaling	6 de noviembre de 2014	Durante una actualización, puede utilizar la marca <code>WaitOnResourceSignals</code> para indicar a CloudFormation que espere a que las instancias envíen una señal de éxito. De ese modo, CloudFormation no actualizará el siguiente lote de instancias hasta que el lote actual esté listo. Para obtener más información, consulte Atributo UpdatePolicy .	2010-05-15
Nuevos atributos VPC <code>Fn::GetAtt</code>	6 de noviembre de 2014	Dado un ID de VPC, puede recuperar el grupo de seguridad predeterminado y una ACL de red para esa VPC. Para obtener más información, consulte Fn::GetAtt .	2010-05-15
Nuevos tipos de parámetros específicos de AWS	6 de noviembre de 2014	Puede especificar los tipos de parámetros específicos de AWS en sus plantillas de CloudFormation. En la consola de CloudFormation, estos tipos de parámetros proporcionan una lista desplegable de valores válidos. Con la API o con AWS CLI, CloudFormation puede validar rápidamente los valores para estos tipos de parámetros antes de crear o actualizar una pila. Para obtener más información, consulte Parámetros .	2010-05-15
Atributo <code>CreationPolicy</code>	6 de noviembre de 2014	Con el atributo <code>CreationPolicy</code> , puede indicar a CloudFormation, que espere hasta que las aplicaciones estén listas para instancias de EC2 antes de proceder con la creación de la pila. Puede utilizar una política de creación en lugar de una condición de espera y un controlador de condición de espera. Para obtener más información, consulte Atributo CreationPolicy .	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Valores reenviados de Amazon CloudFront	29 de septiembre de 2014	Para comportamientos de la caché, puede reenviar los encabezados al origen. Consulte ForwardedValues .	2010-05-15
AWS OpsWorks update	29 de septiembre de 2014	Para Chef 11.10, puede utilizar la propiedad <code>ChefConfiguration</code> para habilitar Berkshelf. También puede utilizar los grupos de seguridad integrados de AWS OpsWorks con sus pilas de AWS OpsWorks. Para obtener más información, consulte AWS::OpsWorks::Stack .	2010-05-15
Soporte del etiquetado de Elastic Load Balancing	29 de septiembre de 2014	Equilibradores de carga de Elastic Load Balancing de etiquetas de AWS CloudFormation con etiquetas de pila. También puede añadir sus propias etiquetas a un balanceador de carga. Consulte AWS::ElasticLoadBalancing::LoadBalancer .	2010-05-15
Actualización de la política del tema Amazon Simple Notification Service	29 de septiembre de 2014	Ahora puede actualizar las políticas de los temas de Amazon SNS. Para obtener más información, consulte AWS::SNS::TopicPolicy .	2010-05-15
Actualización de la instancia de base de datos de RDS	5 de septiembre de 2014	Puede especificar si una instancia de base de datos está expuesta a Internet mediante la propiedad <code>PubliclyAccessible</code> en el recurso AWS::RDS::DBInstance .	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Actualización del atributo UpdatePolicy	05 de septiembre de 2014	Puede especificar una política de actualización para un grupo de Auto Scaling que esté asociado a una acción programada. Para obtener más información, consulte Atributo UpdatePolicy .	2010-05-15
Compatibilidad con Amazon CloudWatch	10 de julio de 2014	Puede usar CloudFormation para aprovisionar y administrar grupos de registro y filtros de métricas de registros de Amazon CloudWatch (registros de CloudWatch). Para obtener más información, consulte AWS::Logs::LogGroup o AWS::Logs::MetricFilter .	2010-05-15
Actualización de configuración de distribución de Amazon CloudFront	17 de junio de 2014	Puede especificar propiedades de configuración de distribución de CloudFront adicionales: • Respuestas de error personalizadas definen mensajes de error personalizados para los códigos de estado HTTP 4xx y 5xx. • La clase de precio define el precio máximo que desea pagar por el servicio de CloudFront. • Las restricciones definen quién puede ver su contenido. • El certificado de usuarios especifica el certificado que se debe usar cuando los espectadores usan HTTPS. • Para comportamientos de la caché, puede especificar los métodos de HTTP permitidos e indicar si reenviar las cookies. Para obtener más información, consulte AWS::CloudFront::Distribution.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Actualización de la instancia de EC2	17 de junio de 2014	Puede especificar que una instancia se detenga o termine al invocar el comando para apagado del sistema operativo de la instancia. Para obtener más información, consulte AWS::EC2::Instance .	2010-05-15
Actualización de volúmenes de EBS	17 de junio de 2014	Puede usar volúmenes de EBS cifrados con tipos de instancias admitidos. Para obtener más información, consulte AWS::EC2::Volume .	2010-05-15
Nueva conexión de emparejamiento de Amazon VPC	17 de junio de 2014	Puede usar CloudFormation para crear una conexión de emparejamiento de Amazon Virtual Private Cloud (Amazon VPC), que establece una conexión de red entre dos VPC. Para obtener más información, consulte AWS::EC2::VPCPeeringConnection .	2010-05-15
Actualización de grupo de Amazon EC2 Auto Scaling	17 de junio de 2014	Puede especificar un grupo de ubicación en clúster existente en el que lanzar instancias para un grupo de Amazon EC2 Auto Scaling. Para obtener más información, consulte AWS::AutoScaling::AutoScalingGroup .	2010-05-15
AWS CloudTrail ICompatibilidad con	17 de junio de 2014	CloudFormation es compatible con AWS CloudTrail, que puede capturar las llamadas a la API hechas desde su cuenta de AWS y publicar los registros en la ubicación que designe. Para obtener más información, consulte AWS::CloudTrail::Trail .	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Actualización de las mejoras de las pilas	12 de mayo de 2014	CloudFormation admite características adicionales para la actualización de pilas: • Puede actualizar los parámetros de pilas de CloudFormation sin necesidad de volver a enviar la plantilla de la pila. • Puede añadir o eliminar temas de notificación de Amazon SNS para una pila de CloudFormation. Para obtener más información, consulte Actualizaciones de pilas de CloudFormation.	2010-05-15
Compatibilidad con Amazon Kinesis	6 de mayo de 2014	Puede usar CloudFormation para crear secuencias de Amazon Kinesis que capturan y transportan los registros de datos desde orígenes de datos. Para obtener más información, consulte AWS::Kinesis::Stream.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Nuevas propiedades del bucket de S3	5 de mayo de 2014	CloudFormation admite las propiedades adicionales del bucket de S3: • El uso compartido de recursos entre orígenes (CORS) define el uso compartido de recursos entre orígenes de objetos en un bucket. • El ciclo de vida define cómo administra Amazon S3 objetos durante su ciclo de vida. • La política de registro de acceso recopila información sobre las solicitudes realizadas a su bucket. • Las notificaciones definen qué eventos informar y a qué tema de Amazon SNS enviar mensajes. • El control de versiones permite múltiples variantes de todos los objetos en un bucket. • Reglas de redireccionamiento y de enrutamiento rigen el comportamiento de redireccionamiento para las solicitudes realizadas a un punto de enlace del sitio web de un bucket. Para obtener más información, consulte AWS::S3::Bucket.	2010-05-15
Compatibilidad con Amazon EC2 Auto Scaling	5 de mayo de 2014	CloudFormation admite la recopilación de métricas para un grupo de escalado automático. Para obtener más información, consulte AWS::AutoScaling::AutoScalingGroup.	2010-05-15
Fn::If update	5 de mayo de 2014	Puede utilizar la función intrínseca Fn::If en la sección de salida de una plantilla. Para obtener más información, consulte Funciones de condición.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Registro de API con AWS CloudTrail	2 de abril de 2014	Puede utilizar AWS CloudTrail (CloudTrail) para registrar solicitudes de CloudFormation. Con CloudTrail puede obtener un historial de las llamadas a la API de CloudFormation para su cuenta. Para obtener más información, consulte Registro de llamadas a la API de CloudFormation con AWS CloudTrail .	2010-05-15
Elastic Load Balancing update	20 de marzo de 2014	Puede especificar una política de registro de acceso para capturar información sobre las solicitudes realizadas a su balanceador de carga. También puede especificar una política de vaciado de conexión que describe cómo administrar las solicitudes en tránsito cuando se cancela el registro de las instancias o están en mal estado. Para obtener más información, consulte AWS::ElasticLoadBalancing::LoadBalancer .	2010-05-15
OpsWorksCompatibilidad con	3 de marzo de 2014	Puede usar CloudFormation para aprovisionar y administrar las pilas de OpsWorks. Para obtener más información, consulte AWS::OpsWorks::Stack o Fragmentos de código de plantillas de AWS OpsWorks .	2010-05-15
Aumento del límite de tamaño de la plantilla de Amazon S3	18 de febrero de 2014	Puede especificar tamaños de plantilla de hasta 460 800 bytes en Amazon S3.	2010-05-15
Compatibilidad con Amazon Redshift	10 de febrero de 2014	Puede usar CloudFormation para aprovisionar y administrar clústeres de Amazon Redshift. Para obtener más información, consulte Fragmentos de plantillas de Amazon Redshift o AWS::Redshift::Cluster .	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Actualización de políticas de buckets y buckets de S3	10 de febrero de 2014	Puede actualizar algunas propiedades de los recursos de la política de buckets y de buckets de S3. Para obtener más información, consulte AWS::S3::Bucket o AWS::S3::BucketPolicy .	2010-05-15
Actualización de versiones de la aplicación y entornos de Elastic Beanstalk	10 de febrero de 2014	Puede actualizar versiones de la aplicación y configuraciones de entorno de Elastic Beanstalk. Para obtener más información, consulte AWS::ElasticBeanstalk::Environment , AWS::ElasticBeanstalk::ConfigurationTemplate o AWS::ElasticBeanstalk::ApplicationVersion .	2010-05-15
Amazon SQS update	29 de enero de 2014	Puede especificar una cola de mensajes fallidos de una cola de Amazon SQS. Para obtener más información, consulte AWS::SQS::Queue .	2010-05-15
Acciones programadas de Auto Scaling	27 de enero de 2014	Puede escalar el número de instancias de EC2 en un grupo de Auto Scaling en función de un calendario. Al utilizar un calendario, puede escalar aplicaciones en respuesta a cambios de carga predecibles. Para obtener más información, consulte AWS::AutoScaling::ScheduledAction .	2010-05-15
Índices secundarios de DynamoDB	27 de enero de 2014	Puede crear índices secundarios locales y globales para bases de datos DynamoDB. Al utilizar índices secundarios, puede obtener acceso de forma eficaz a los datos con atributos distintos de la clave principal. Para obtener más información, consulte AWS::DynamoDB::Table .	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Auto Scaling update	2 de enero de 2014	Puede especificar un ID de instancia de un grupo de Auto Scaling o configuración de lanzamiento. También puede especificar las propiedades de dispositivos de bloques de Auto Scaling adicionales. Para obtener más información, consulte AWS::AutoScaling::AutoScalingGroup o AWS::AutoScaling::LaunchConfiguration .	2010-05-15
Amazon SQS update	2 de enero de 2014	Puede actualizar colas de SQS y especificar propiedad es adicionales. Para obtener más información, consulte AWS::SQS::Queue .	2010-05-15
Aumentos de límites	2 de enero de 2014	Puede especificar hasta 60 parámetros y 60 salidas en sus plantillas de CloudFormation.	2010-05-15
Nueva consola	19 de diciembre de 2013	La nueva consola de CloudFormation agrega características como eventos de pila que se actualizan automáticamente y ordenación alfabética de los parámetros de la pila.	2010-05-15
Balance de carga entre zonas	19 de diciembre de 2013	Con el equilibrio de carga entre zonas, puede dirigir el tráfico a instancias del back-end entre todas las zonas de disponibilidad (AZ). Para obtener más información, consulte AWS::ElasticLoadBalancing::LoadBalancer .	2010-05-15
AWS Elastic Beanstalk Capas del entorno de	19 de diciembre de 2013	Puede especificar si AWS Elastic Beanstalk aprovisiona recursos para dar soporte a un servidor web o para administrar tareas de procesamiento de fondo. Para obtener más información, consulte AWS::ElasticBeanstalk::Environment .	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Nombres de recursos	19 de diciembre de 2013	Puede asignar nombres (ID físico) para los siguientes recursos: • Clústeres de ElastiCache • Balanceadores de carga de Elastic Load Balancing • Instancias de base de datos de RDS Para obtener más información, consulte Tipo de nombre.	2010-05-15
Compatibilidad con VPN	22 de noviembre de 2013	Puede habilitar una gateway privada virtual (VGW) para propagar rutas a las tablas de ruteo de una VPC. Para obtener más información, consulte AWS::EC2::VPNGatewayRoutePropagation.	2010-05-15
Creación de recursos y asignación de propiedades en función de condiciones	8 de noviembre de 2013	Puede utilizar parámetros de entrada para controlar la creación y los ajustes de los recursos de la pila designada, definiendo las condiciones en sus plantillas de CloudFormation. Por ejemplo, puede utilizar condiciones para crear los recursos de la pila de un entorno de producción. Con la misma plantilla, puede crear recursos de pila similares con menor capacidad para un entorno de prueba. Para obtener más información, consulte Funciones de condición.	2010-05-15
Prevención de actualizaciones involuntarias de los recursos de la pila	8 de noviembre de 2013	Puede impedir actualizaciones de la pila que podrían provocar cambios involuntarios en los recursos de la pila. Por ejemplo, si tiene una pila con una capa de la base de datos que apenas se debe actualizar, puede definir una política de pila que impida que la mayoría de los usuarios actualicen esa capa de la base de datos. Para obtener más información, consulte Evitar actualizaciones en los recursos de la pila.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Denominación de recursos	8 de noviembre de 2013	En lugar de utilizar ID físicos generados por CloudFormation, puede asignar nombres a determinados recursos. Los siguientes recursos de CloudFormation admiten la asignación de nombres • Alarmas de Amazon CloudWatch • Tablas de DynamoDB • Aplicaciones y entornos de AWS Elastic Beanstalk • Buckets de Amazon S3 • Temas de Amazon SNS • Colas de Amazon SQS Para obtener más información, consulte Tipo de nombre.	2010-05-15
Asignación de tipos de recurso personalizados	8 de noviembre de 2013	En las plantillas, puede especificar su propio tipo de recurso para los recursos personalizados de CloudFormation (<code>AWS::CloudFormation::CustomResource</code>). Al utilizar su propio nombre de tipo de recurso personalizado, puede identificar rápidamente el tipo de recursos personalizados que tiene en su pila. Por ejemplo, puede especificar "Type": "<code>Custom::MyCustomResource</code>". Para obtener más información, consulte AWS::CloudFormation::CustomResource.	2010-05-15
Adición de pseudoparámetro	8 de noviembre de 2013	Ahora puede consultar el AccountID de AWS dentro de las plantillas de CloudFormation consultando el pseudoparámetro <code>AWS::AccountId</code>. Para obtener más información, consulte Referencia de pseudoparámetros.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Especificación de pilas en las políticas de IAM	8 de noviembre de 2013	Puede permitir o denegar a grupos, usuarios o roles de IAM que operen en pilas de CloudFormation específicas. Por ejemplo, puede denegar la acción de eliminación de pila en un ID de pila específico. Para obtener más información, consulte Controlar el acceso con AWS Identity and Access Management .	2010-05-15
Soporte para federación	14 de octubre de 2013	CloudFormation admite credenciales de seguridad temporales de los roles de IAM, que permiten escenarios como la federación y el inicio de sesión único en la Consola de administración de AWS. También puede realizar llamadas a CloudFormation desde las instancias de EC2 sin incrustar credenciales de seguridad a largo plazo mediante la utilización de roles de IAM. Para obtener más información acerca de IAM y CloudFormation, consulte Controlar el acceso con AWS Identity and Access Management .	2010-05-15
Soporte para réplica de lectura de Amazon RDS	24 de septiembre de 2013	Ahora puede crear réplicas de lectura de Amazon RDS a partir de una instancia de base de datos de origen. Para obtener más información, consulte la propiedad <code>SourceDBInstanceIdentifier</code> en el recurso AWS::RDS::DBInstance .	2010-05-15
Asociación de dirección IP pública a instancias en un grupo de Auto Scaling	19 de septiembre de 2013	Ahora puede asociar direcciones IP públicas con instancias en un grupo de Auto Scaling. Para obtener más información, consulte AWS::AutoScaling::LaunchConfiguration .	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Soporte para VPC adicional	17 de septiembre de 2013	CloudFormation añade varias mejoras para admitir VPC y la funcionalidad VPN • Puede asociar una dirección IP pública y varias direcciones IP privadas a interfaces de red Amazon EC2. Para obtener más información, consulte AWS::EC2::NetworkInterface. También puede asociar una dirección IP privada principal a una dirección IP elástica (EIP). • Puede habilitar la compatibilidad de DNS y especificar nombres de host DNS. Para obtener más información, consulte AWS::EC2::VPC. • Puede especificar una ruta estática entre una gateway privada virtual y su gateway de VPN. Para obtener más información, consulte AWS::EC2::VPNConnectionRoute.	2010-05-15
Soporte para Redis y grupos de seguridad de VPC para Amazon ElastiCache	3 de septiembre de 2013	Ahora puede especificar Redis como el motor de almacenamiento en caché para un clúster de Amazon ElastiCache (ElastiCache). Ahora también puede asignar grupos de seguridad de VPC para los clústeres de ElastiCache. Para obtener más información, consulte AWS::ElastiCache::CacheCluster.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Creación, actualización y eliminación de la pila paralela y actualizaciones de pilas anidadas	12 de agosto de 2013	AWS CloudFormation crea, actualiza y elimina ahora recursos en paralelo, lo que mejora el desempeño de las operaciones. Si actualiza una plantilla de nivel superior, AWS CloudFormation actualiza automáticamente pilas anidadas que han cambiado. Para obtener más información, consulte Actualizaciones de pilas de CloudFormation .	2010-05-15
Configuración de grupos de seguridad de VPC en instancias de base de datos de RDS	28 de febrero de 2013	Ahora puede asignar grupos de seguridad de VPC para una instancia de base de datos de RDS con CloudFormation. Para obtener más información, consulte la propiedad VPCSecurityGroups en AWS::RDS::DBInstance .	2010-05-15
Implementaciones continuas para grupos de Amazon EC2 Auto Scaling	20 de febrero de 2013	CloudFormation ahora es compatible con las políticas de actualización de los grupos de Amazon EC2 Auto Scaling, que describen el modo en que se sustituyen o modifican las instancias en el grupo de Amazon EC2 Auto Scaling cuando el grupo de Amazon EC2 Auto Scaling añade o elimina instancias. Puede modificar estos ajustes al crear la pila o durante la actualización de una pila. Para obtener más información y un ejemplo, consulte Atributo UpdatePolicy .	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Cancelación y acción de restauración para actualizaciones de pila	20 de febrero de 2013	CloudFormation admite la posibilidad de cancelar las actualizaciones de una pila. La pila debe estar en el estado UPDATE_IN_PROGRESS cuando se realice la solicitud de actualización. Hay más información disponible en los siguientes temas: • Cancelación de una actualización de pila • cancel-update-stack • CancelUpdateStack en la Referencia de la API de CloudFormation	2010-05-15
Instancias optimizadas para EBS para grupos de Amazon EC2 Auto Scaling	20 de febrero de 2013	Ahora puede aprovisionar instancias optimizadas para EBS en grupos de Amazon EC2 Auto Scaling para una capacidad de proceso dedicada para Amazon Elastic Block Store (Amazon EBS) en instancias de escalado automático. La implementación es similar a la de la compatibilidad publicada anteriormente optimizadas para las instancias de EC2 de Amazon EBS. Para obtener más información, consulte la nueva propiedad EbsOptimized en AWS::AutoScaling::LaunchConfiguration.	2010-05-15
Nueva documentación	21 de diciembre de 2012	AWS::EC2::Instance ofrece ahora una propiedad BlockDeviceMappings que le permite asignaciones de dispositivos de bloques para la instancia EC2. Con este cambio, se agregaron dos tipos nuevos: • BlockDeviceMapping • Dispositivo de bloques	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Nueva documentación	21 de diciembre de 2012	Se han agregado nuevas secciones para describir los procedimientos para la creación y visualización de pilas mediante la recién rediseñada Consola de administración de AWS. Puede encontrarlos aquí: • Creación de una pila en la consola de CloudFormation • Visualización de la información de la pila desde la consola de CloudFormation	2010-05-15
Nueva documentación	15 de noviembre de 2012	Se proporciona información sobre recursos personalizados en los siguientes temas: • Recursos personalizados • AWS::CloudFormation::CustomResource • Referencia de recursos personalizados	2010-05-15
Documentación actualizada	15 de noviembre de 2012	CloudFormation ahora es compatible con la especificación de operaciones de E/S por segundo (IOPS) aprovisionadas para instancias de base de datos de RDS. Puede definir este valor de 1000 a 10 000 en incrementos de 1000 IOPS a través de la nueva propiedad IOPS en AWS::RDS::DBInstance. Para obtener más información sobre la especificación de IOPS para instancias de base de datos de RDS, consulte IOPS provisionadas en la Guía del usuario de Amazon Relational Database Service.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Documentación nueva y actualizada	27 de agosto de 2012	Se han reorganizado los temas para proporcionar información específica de manera más clara sobre el uso de la Consola de administración de AWS y el uso de la interfaz de la línea de comandos (CLI) de CloudFormation. Se añadió información sobre el etiquetado de pilas de CloudFormation, incluidas guías nuevas y temas de referencia actualizados: • Nuevo tema: Configurar las opciones de pila. • Nueva información sobre las etiquetas en la Referencia de la API de CloudFormation: CreateStack, Stack y Tag. Nueva información sobre cómo trabajar con pilas de Windows: • Plantillas de CloudFormation e imágenes de máquina de Amazon (AMI) de Microsoft Windows • Arranque de pilas de Windows de CloudFormation Nuevo tema: Uso de expresiones regulares en plantillas de CloudFormation.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Nueva característica	25 de abril de 2012	CloudFormation ahora ofrece compatibilidad completa de seguridad para la nube privada virtual (VPC) con Amazon EC2. Ahora puede crear y rellenar toda una VPC con cada tipo de recurso de VPC (subredes, puertas de enlace, ACL de redes, tablas de enrutamiento, etc.) con una única plantilla de CloudFormation. Se ha actualizado la documentación para los siguientes tipos de recursos: • AWS::EC2::SecurityGroup • AWS::EC2::SecurityGroupIngress • AWS::EC2::SecurityGroupEgress • AWS::EC2::Instance • AWS::AutoScaling::AutoScalingGroup • AWS::EC2::EIP • AWS::EC2::EIPAssociation • AWS::ElasticLoadBalancing::LoadBalancer Adición de nuevos tipos de recursos a la documentación: • AWS::EC2::VPC • AWS::EC2::InternetGateway • AWS::EC2::DHCPOptions • AWS::EC2::DHCPOptions • AWS::EC2::RouteTable • AWS::EC2::NetworkAcl • AWS::EC2::NetworkAclEntry • AWS::EC2::Subnet • AWS::EC2::VPNGateway	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
		• AWS::EC2::CustomerGateway	
Nueva característica	13 de abril de 2012	CloudFormation ahora le permite añadir o eliminar elementos de una pila al actualizarla. Actualizaciones de pilas de CloudFormation se ha actualizado y se ha agregado una nueva sección al tutorial Cambiar los recursos de la pila , que describe cómo agregar y eliminar recursos al actualizar la pila.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Nueva característica	2 de febrero de 2012	CloudFormation ahora proporciona soporte para recursos en una Amazon Virtual Private Cloud (Amazon VPC) existente. En esta versión, puede: - Lanzar una instancia de EC2 dedicada en una Amazon VPC existente. Para obtener más información, consulte AWS::EC2::Instance. - Establezca el atributo <code>SourceDestCheck</code> de una instancia de EC2 que reside en una Amazon VPC existente. Para obtener más información, consulte AWS::EC2::Instance. - Crear direcciones IP elásticas en un dominio existente de Amazon VPC. Para obtener más información, consulte AWS::EC2::EIP. - Utilice CloudFormation para crear grupos de seguridad de Amazon VPC y las reglas de entrada/salida en una VPC existente. Para obtener más información, consulte AWS::EC2::SecurityGroup. - Asocie un grupo de Auto Scaling con una Amazon VPC existente mediante la configuración de la propiedad <code>VPCZoneIdentifier</code> de su recurso <code>AWS::AutoScaling::AutoScalingGroup</code>. Para obtener más información, consulte AWS::AutoScaling::AutoScalingGroup. - Asocie un balanceador de carga de Elastic Load Balancing a una subred de Amazon VPC y cree grupos de seguridad para el balanceador de carga. Para obtener más información, consulte AWS::ElasticLoadBalancing::LoadBalancer.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
		• Crear una instancia de base de datos de RDS en una Amazon VPC existente. Para obtener más información, consulte AWS::RDS::DBInstance.	
Nueva característica	2 de febrero de 2012	Ahora puede actualizar propiedades para los siguientes recursos en una pila existente: • AWS::EC2::SecurityGroupIngress • AWS::EC2::SecurityGroupEgress • AWS::EC2::EIPAssociation • AWS::RDS::DBSubnetGroup • AWS::RDS::DBSecurityGroup • AWS::RDS::DBSecurityGroupIngress • AWS::Route53::RecordSetGroup Para obtener una lista completa de los recursos que se pueden actualizar y detalles sobre lo que hay que tener en cuenta al actualizar una pila, consulte Actualizaciones de pilas de CloudFormation.	2010-05-15
Guía reestructurada	2 de febrero de 2012	Secciones existentes reorganizadas en nuevas secciones: Trabajar con plantillas de CloudFormation y Administración de pilas. La Referencia de plantilla se ha pasado al nivel superior del Índice Se ha desplazado Estimación del costo de su pila de CloudFormation a la sección de introducción.	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Nuevo contenido	2 de febrero de 2012	Se agregaron nuevas secciones: • Tutorial: Actualización de una pila es un tutorial que recorre el proceso de actualización de una pila LAMP. • Implementación de aplicaciones en Amazon EC2 con AWS CloudFormation describe cómo utilizar scripts auxiliares de CloudFormation para implementar aplicaciones con los metadatos almacenados en su plantilla. • Referencia de scripts auxiliares de CloudFormation proporciona material de referencia para los scripts auxiliares de CloudFormation (cfn-init, cfn-get-metadata, cfn-signal y cfn-hup).	2010-05-15
Nueva característica	26 de mayo de 2011	CloudFormation proporciona ahora el comando <code>list-stacks</code> que le permite enumerar pilas filtradas en función del estado de la pila. Las pilas eliminadas pueden enumerarse hasta 90 días después de que se hayan eliminado. Para obtener más información, consulte Descripción y publicación de pilas .	2010-05-15
Nuevas características	26 de mayo de 2011	Los comandos <code>describe-stack-resources</code> y <code>get-template</code> le permiten ahora obtener información de pilas que se han eliminado durante 90 días después de que se hayan eliminado. Para obtener más información, consulte Listado de recursos y Recuperación de una plantilla .	2010-05-15
Nuevo enlace	1 de marzo de 2011	La información de punto de conexión de CloudFormation se encuentra en Referencia general de AWS. Para obtener más información, vaya a Regiones y puntos de enlace en la Referencia general de AWS .	2010-05-15

Cambio	Fecha de lanzamiento	Descripción	Versión de API
Versión inicial	25 de febrero de 2011	La versión pública inicial de CloudFormation.	2010-05-15