



Guía para desarrolladores

AWS Serverless Application Model



AWS Serverless Application Model: Guía para desarrolladores

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 AWS SAM?	1
Cuándo se debe usar AWS SAM	1
Características principales de	2
Información relacionada	3
Funcionamiento	3
¿Qué es el AWS SAM ¿especificación de plantilla?	4
¿Qué es el AWS SAM proyecto y AWS SAM plantilla?	4
¿Qué es el AWS SAM CLI?	11
Más información	19
Siguientes pasos	19
Conceptos sobre sistemas sin servidor	19
Conceptos sobre sistemas sin servidor	20
Infraestructura como código	21
Herramientas de IaC para AWS aplicaciones sin servidor	21
Comparación de herramientas de IaC para aplicaciones sin servidor	22
Más información	23
Introducción	24
Requisitos previos	24
Inscríbase en una Cuenta de AWS	25
Paso 1: Instale el AWS CLI	25
Paso 2: utilice el AWS CLI para configurar AWS credentials	25
Paso 3: Instalar (opcional) AWS Kit de herramientas para VS Code	26
Siguientes pasos	27
Instala la AWS SAM CLI	27
Instalación de la AWS SAM CLI	28
Solución de errores de la instalación	38
Siguientes pasos	40
Opcional: Comprobación del instalador de la CLI de AWS SAM	40
Tutorial Hello World	53
Requisitos previos	54
Paso 1: inicializar la aplicación de muestra Hello World	55
Paso 2: crear tu propia aplicación	59
Paso 3: Implemente la aplicación en Nube de AWS	61
Paso 4: ejecutar la aplicación	66

Paso 5: Interactúa con tu función en el Nube de AWS	67
Paso 6: Modifique y sincronice su aplicación con Nube de AWS	68
Paso 7: (opcional) probar tu aplicación localmente	71
Paso 8: Elimine la aplicación del Nube de AWS	73
Resolución de problemas	74
Más información	74
Conversión desde Lambda	74
Más información	75
Cómo usar AWS SAM	76
AWS SAM CLI	77
Cómo AWS SAM CLI los comandos están documentados	77
Configuración de la AWS SAM de CLI	79
Cómo crear el archivo de configuración (el archivo <code>samconfig</code>)	80
Configurar los ajustes del proyecto	80
Configura las credenciales y los ajustes básicos	85
Comandos principales	85
Pruebas locales con la de AWS SAM CLI	86
Test AWS SAM aplicaciones a nivel local	87
Test CloudFormation plantillas de forma local	87
Test AWS Aplicaciones CDK a nivel local	87
Probar las aplicaciones de Terraform de forma local	88
AWS SAM plantilla	89
Anatomía de la plantilla	90
YAML	91
Secciones de la plantilla	91
Pasos a seguir a continuación	94
Globals	94
Recursos y propiedades	100
AWS::Serverless: :API	101
AWS::Serverless::Application	162
AWS::Serverless::CapacityProvider	167
AWS::Serverless: :Conector	179
AWS::Serverless: :Función	193
AWS::Serverless:: API GraphQL	360
AWS::Serverless::HttpApi	396
AWS::Serverless::WebSocketApi	428

AWS::Serverless::LayerVersion	448
AWS::Serverless::MicrovmImage	455
AWS::Serverless::NetworkConnector	462
AWS::Serverless::SimpleTable	469
AWS::Serverless::StateMachine	475
AWS::Serverless::WebFunction	517
Recursos de generados	536
Referencia generada CloudFormation recursos	536
Generado CloudFormation escenarios de recursos	537
AWS::Serverless::Api	541
AWS::Serverless::Application	543
AWS::Serverless::CapacityProvider	543
AWS::Serverless::Connector	544
AWS::Serverless::Function	545
AWS::Serverless::GraphQLApi	551
AWS::Serverless::HttpApi	553
AWS::Serverless::MicrovmImage	554
AWS::Serverless::NetworkConnector	555
AWS::Serverless::WebFunction	556
AWS::Serverless::WebSocketApi	558
AWS::Serverless::LayerVersion	561
AWS::Serverless::SimpleTable	561
AWS::Serverless::StateMachine	561
Atributos de recursos compatibles	563
Excepciones	564
Extensiones de API Gateway	565
Funciones intrínsecas	566
Soporte de extensiones de idioma	567
Habilitar las extensiones de idioma	568
Ventilador:: ForEach	570
Funciones intrínsecas compatibles	577
Errores de validación	578
Limitaciones	579
Telemetría	580
Ejemplo de aplicaciones	581
Proceso de eventos de DynamoDB	581

Antes de empezar	581
Paso 1: Inicializar la aplicación	581
Paso 2: probar la aplicación localmente	582
Paso 3: crear el paquete de la aplicación	582
Paso 4: implementación de la aplicación	583
Siguientes pasos	584
Procesar eventos de Amazon S3	584
Antes de empezar	584
Paso 1: Inicializar la aplicación	585
Paso 2: Crear el paquete de la aplicación	585
Paso 3: implementar de la aplicación	586
Paso 4: probar la aplicación de forma local	587
Próximos pasos a seguir	587
Desarrolle su la aplicación	588
Creación de una aplicación	588
Inicialización de aplicación sin servidor	589
Opciones para el inicio de sesión	595
Resolución de problemas	595
Ejemplos	596
Más información	596
Siguientes pasos	597
Definición de la infraestructura	597
Definición de los recursos de la aplicación	597
Configuración de acceso	599
Control del acceso a la API	679
Aumento de la eficiencia con capas	692
Reutilización del código	695
Administración de eventos basados en tiempo	698
Orquestación de aplicaciones	702
Configuración de la firma de código	703
Validar archivos AWS SAM de plantilla	707
Creación de su aplicación	707
Introducción a sam build	708
Compilación predeterminada	723
Personalización de su compilación	732
Prueba de la aplicación	766

Introducción a sam local	766
Uso de sam local comando	767
Introducción a sam local generate-event	767
Introducción a sam local invoke	774
Introducción a sam local start-api	780
Introducción a sam local start-lambda	787
Invocación local de funciones	790
Archivo de variables de entorno	791
Capas	793
Más información	793
Ejecución local de API Gateway	793
Archivo de variables de entorno	795
Capas	797
Pruebas con sam remote test-event	797
Configure el AWS SAM CLI para usar evento de prueba remoto de sam	798
Uso de evento de prueba remoto de sam comando	798
Uso de eventos de prueba compartibles	801
Gestion de eventos de prueba compartibles	801
Pruebas con sam remote invoke	803
Uso del comando sam remote invoke	804
Uso de las opciones del comando sam remote invoke	808
Configura el archivo de configuración de su proyecto	814
Ejemplos	814
Vínculos relacionados	830
Automatización de las pruebas de integración	830
Generación de cargas útiles de ejemplo	832
Pruebe y depure funciones duraderas	833
Flujo de trabajo de pruebas locales	833
Escenarios de prueba comunes	833
Resolución de problemas	834
Consejos de depuración	835
Más información	835
Depuración de la aplicación	836
Depuración local de funciones	836
Utilización AWS Kits de herramientas	837
Ejecutar AWS SAM localmente en modo de depuración	839

Envío de varios argumentos de tiempo de ejecución	840
Valide con cfn-lint	840
Ejemplos	841
Más información	841
Implementación de la aplicación y los recursos	842
Introducción a sam deploy	842
Requisitos previos	843
Implementación de aplicaciones mediante sam deploy	843
Prácticas recomendadas	854
Opciones para una implementación de sam	854
Resolución de problemas	854
Ejemplos	854
Más información	863
Opciones de implementación	863
Cómo usar la CLI de AWS SAM para realizar una implementación manual	863
Implemente con CI/CD sistemas y canalizaciones	864
Implementación gradual	864
Solución de problemas de las implementaciones mediante la CLI de AWS SAM	864
Más información	793
Implemente con CI/CD sistemas y canalizaciones	865
¿Qué es una canalización?	866
¿Cómo AWS SAM carga los archivos locales	867
Generación de una canalización de inicio	875
Personalización de las canalizaciones de inicio	881
Automatización de las implementaciones	883
Uso de la autenticación de OIDC	888
Introducción a sam sync	890
Detecte y sincronice automáticamente los cambios locales en el Nube de AWS	891
Personalice los cambios locales que se sincronizan con el Nube de AWS	893
Preparar tu aplicación en la nube para probarla y validarla.	893
Opciones para el comando de sincronización sam	893
Resolución de problemas	896
Ejemplos	896
Más información	903
Supervisión de la aplicación	904
Perspectivas de aplicación	904

Configurar CloudWatch Application Insights con AWS SAM	904
Pasos a seguir a continuación	908
Trabajo con registros	908
Obtención de registros mediante CloudFormation pila	908
Obtención de registros por nombre de función de eventos de Lambda	909
Rastreo de registros	909
Visualización de registros para un intervalo de tiempo específico	909
Filtrado de registros	909
Resaltado de errores	909
Impresión clara de código JSON	909
AWS SAM Referencia de	911
AWS SAM la especificación y el AWS SAM plantilla	911
AWS SAM CLI Referencia de los comandos	911
AWS SAM Plantillas de política de	912
Temas	912
Comandos de la CLI de AWS SAM	913
compilación de sam	914
sam delete	920
sam deploy	922
sam init	929
sam list	933
sam local callback	942
sam local execution	948
sam local generate-event	954
invocar sam local	955
sam local start-api	962
sam local start-lambda	969
registros de spam	974
sam package	978
sam pipeline bootstrap	982
sam pipeline init	987
sam publish	988
sam remote callback	990
sam remote execution	997
invocación remota de sam	1004
evento de prueba remoto de sam	1009

sam sync	1016
sam traces	1024
sam validate	1025
Administración de la CLI de AWS SAM	1027
Archivo de configuración de la CLI de AWS SAM	1027
Administración de versiones de AWS SAM CLI	1035
Configuración de AWS credenciales	1044
Telemetría AWS SAM CLI	1046
Resolución de problemas	1048
Referencia de conector	1054
Tipos de recursos de conector admitidos	1054
Políticas de IAM creadas por conectores	1064
Instalar Docker	1088
Instalación Docker	1088
Siguiendo los pasos	1091
Instalación de Finch	1092
Instalación Finch	1092
Configuración Finch por AWS SAM CLI	1094
Verificar la instalación	1095
Detención Finch	1097
Resolución de problemas	1097
Siguiendo los pasos	1098
Repositorios de imágenes	1098
URI del repositorio de imágenes	1099
Ejemplos	1101
Implementación gradual	1101
Implementación gradual de una función de eventos de Lambda por primera vez	1105
Más información	1106
Notas importantes	1106
2023	1106
Compatibilidad con Terraform	1108
Introducción	1108
Requisitos previos	1108
Utilización AWS SAM CLI comandos con Terraform	1109
Configurado para Terraform proyectos	1109
Configurado para Terraform Cloud	1115

Uso de la CLI de AWS SAM con Terraform	1117
Pruebas locales con la de invocar sam local	1117
Pruebas locales con la de sam local start-api	1117
Pruebas locales con la de sam local start-lambda	1119
Terraform limitaciones	1120
Uso de la CLI de AWS SAM con Serverless.tf	1120
Referencia de Terraform	1121
AWS SAM referencia de funciones compatibles	1121
Terraform referencia específica	1121
metadatos de sam	1121
Compatibilidad de Terraform con la CLI de AWS SAM	1124
¿Qué es el AWS SAM CLI?	1125
¿Cómo utilizo el AWS SAM CLI por Terraform?	1126
Pasos a seguir a continuación	1126
Publicación para que otros usuarios la usen	1127
Requisitos previos	1127
Publicación de una aplicación	1129
Paso 1: Añada una sección de metadatos a AWS SAM plantilla	1129
Paso 2: crear el paquete de la aplicación	1129
Paso 3: publicar la aplicación	1130
Paso 4: compartir la aplicación (opcional)	1130
Publicar una nueva versión de una aplicación existente	1131
Temas adicionales	1131
Propiedades de la sección metadatos	1131
Propiedades	1131
Casos de uso	1134
Ejemplo	1135
Historial de revisión	1137
.....	mclxxviii

¿Qué es el AWS Serverless Application Model (AWS SAM)?

AWS Serverless Application Model (AWS SAM) es un marco de código abierto para crear aplicaciones sin servidor utilizando la infraestructura como código (IaC). Con AWS SAM su sintaxis abreviada, los desarrolladores declaran los recursos y los [CloudFormation](#) recursos especializados sin servidor que se transforman en infraestructura durante la implementación. Al trabajar con AWS SAM, interactuará con:

1. AWS SAM CLI- Una herramienta de línea de comandos que le ayuda a desarrollar, probar e implementar localmente sus aplicaciones sin servidor.
2. AWS SAM Plantilla: una extensión CloudFormation que proporciona una sintaxis simplificada para definir los recursos sin servidor.

Al utilizar el `sam init` comando, se crea un directorio de proyectos, al que llamaremos AWS SAM proyecto, que normalmente incluye la AWS SAM plantilla, el código de la aplicación y otros archivos de configuración.

Cuándo se debe usar AWS SAM

AWS SAM es una solución de IaC ideal para escenarios en los que desee un desarrollo simplificado sin servidor con toda la potencia de CloudFormation. Por ejemplo, puede utilizar SAM para los siguientes procesos:

- Aplicaciones sin servidor: puede usar SAM para definir rápidamente funciones de AWS Lambda eventos (anteriormente denominadas funciones Lambda), funciones duraderas de Lambda, API de Amazon API Gateway, tablas de Amazon DynamoDB y otros recursos sin servidor con un código mínimo.
- CloudFormation mejora: puede combinar SAM con las CloudFormation plantillas existentes para añadir componentes sin servidor a la infraestructura tradicional. Los recursos de SAM funcionan junto con CloudFormation los recursos estándar en la misma plantilla.
- Desarrollo y pruebas locales: puede utilizar la CLI de SAM para probar las funciones de Lambda a nivel local, así como simular los puntos de conexión de API Gateway y depurar aplicaciones sin servidor en su máquina de desarrollo antes de implementarlas en AWS.
- CI/CD para entornos sin servidor: puede crear canalizaciones de despliegue mediante plantillas SAM que generen automáticamente la CloudFormation infraestructura necesaria para los entornos de ensayo y producción.

- Migración desde recursos creados en la consola: puede convertir las funciones de Lambda y los recursos de API Gateway creados en la consola de administración de AWS en infraestructura como código mediante plantillas de SAM.

Comparación con otras herramientas de AWS SAM IaC

- Utilice SAM en lugar de simplificar las definiciones de recursos sin servidor y, CloudFormation al mismo tiempo, mantener la compatibilidad de las plantillas.
- Utilice SAM en lugar de AWS CDK utilizar un enfoque declarativo para describir su infraestructura en lugar de uno programático.
- Combine SAM con las funciones AWS CDK de pruebas locales de la CLI de SAM para mejorar sus aplicaciones de CDK.

Características principales de

AWS SAM ofrece una variedad de beneficios que mejoran la experiencia del desarrollador al permitirle:

Define el código de infraestructura de tu aplicación rápidamente, utilizando menos código

Cree AWS SAM plantillas para definir el código de infraestructura de aplicaciones sin servidor. Implemente sus plantillas directamente CloudFormation para aprovisionar sus recursos.

Administra las aplicaciones sin servidor durante todo su ciclo de vida de desarrollo

Usa la CLI de AWS SAM para administrar tu aplicación sin servidor durante las fases de autoría, creación, implementación, prueba y supervisión de su ciclo de vida de desarrollo. Para obtener más información, consulte [AWS SAM CLI](#).

Aprovisione rápidamente los permisos entre los recursos con AWS SAM conectores

Usa AWS SAM conectores en tus AWS SAM plantillas para definir los permisos entre tus AWS recursos. AWS SAM transforma tu código en los permisos de IAM necesarios para facilitar tu intención. Para obtener más información, consulte [Administrar los permisos de recursos con AWS SAM conectores](#).

Sincroniza de forma continua los cambios locales en la nube a medida que ve desarrollando contenido

Utilice el AWS SAMCLI `sam sync` comando para sincronizar automáticamente los cambios locales con la nube, lo que agiliza los flujos de trabajo de desarrollo y pruebas en la nube. Para obtener más información, consulta [Introducción al uso sam sync sincronizar con Nube de AWS](#).

Gestione sus aplicaciones de Terraform sin servidor

Use la CLI de AWS SAM para realizar la depuración y las pruebas locales de las funciones y capas de Lambda. Para obtener más información, consulta [AWS SAM CLI Terraform compatibilidad](#).

Información relacionada

- Para obtener información sobre cómo AWS SAM funciona, consulte [Cómo AWS SAM funciona](#).
- Para empezar a usarlo AWS SAM, consulte [Empezar con AWS SAM](#).
- Para obtener información general sobre cómo se puede utilizar AWS SAM para crear una aplicación sin servidor, consulte [Cómo usar AWS SAM](#).

Cómo AWS SAM funciona

Cuando trabaje AWS SAM para crear su aplicación sin servidor, interactuará con los siguientes componentes:

1. [AWS SAM plantilla](#)— Un archivo importante que define sus AWS recursos. Esta plantilla incluye la especificación de la plantilla de AWS SAM : el marco de código abierto que viene con una sintaxis abreviada simplificada que se utiliza para definir las funciones, los eventos, las API, las configuraciones y los permisos de la aplicación sin servidor. Este archivo se encuentra en el AWS SAM proyecto, que es la carpeta de la aplicación que se crea al ejecutar el `sam init` comando.
2. [AWS SAM CLI](#)— Una herramienta de línea de comandos que puede usar con su AWS SAM proyecto y que admite integraciones de terceros para crear y ejecutar sus aplicaciones sin servidor. AWS SAMCLIEs la herramienta que utilizas para ejecutar comandos en tu AWS SAM proyecto y, finalmente, convertirlo en tu aplicación sin servidor.

Para expresar los recursos, las asignaciones de fuentes de eventos y otras propiedades que definen una aplicación sin servidor, defina los recursos y desarrolle la aplicación en la AWS SAM plantilla y

en otros archivos del proyecto. AWS SAM Los usa AWS SAMCLI para ejecutar los comandos en su AWS SAM proyecto, que es la forma de inicializar, compilar, probar e implementar su aplicación sin servidor.

 ¿La primera vez que utiliza la tecnología sin servidor?

También le recomendamos que revise [Conceptos sin servidor para AWS Serverless Application Model](#).

¿Qué es el AWS SAM ¿especificación de plantilla?

La especificación AWS SAM de la plantilla es un marco de código abierto que puede utilizar para definir y administrar el código de infraestructura de aplicaciones sin servidor. La especificación de la AWS SAM plantilla es:

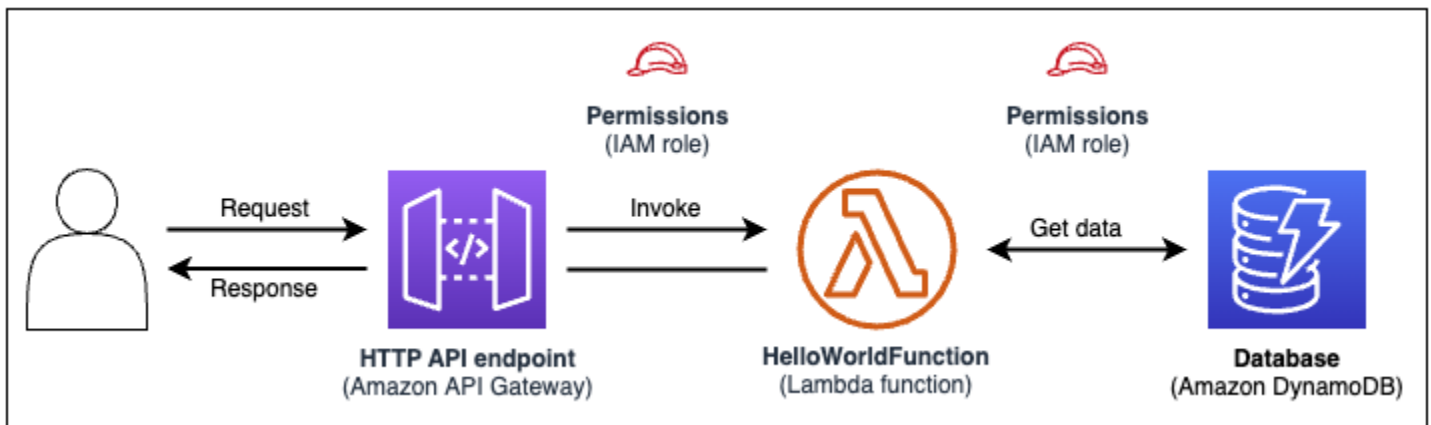
- Basado en AWS CloudFormation: la CloudFormation sintaxis se utiliza directamente en la AWS SAM plantilla y se aprovecha su amplia compatibilidad con las configuraciones de recursos y propiedades. Si ya lo conoce CloudFormation, no necesita aprender un nuevo servicio para administrar el código de infraestructura de sus aplicaciones.
- Una extensión de CloudFormation: AWS SAM ofrece su propia sintaxis única que se centra específicamente en acelerar el desarrollo sin servidores. Puede usar la AWS SAM sintaxis CloudFormation y la sintaxis en la misma plantilla.
- Una sintaxis abstracta y abreviada: con la sintaxis de AWS SAM , puedes definir su infraestructura rápidamente, en menos líneas de código y con una menor probabilidad de errores. Su sintaxis está especialmente diseñada para eliminar la complejidad que implica definir la infraestructura de aplicaciones sin servidor.
- Transformacional: AWS SAM realiza la compleja tarea de transformar la plantilla en el código necesario para aprovisionar la infraestructura. CloudFormation

¿Qué es el AWS SAM proyecto y AWS SAM plantilla?

El AWS SAM proyecto incluye la AWS SAM plantilla que contiene la especificación AWS SAM de la plantilla. Esta especificación es el marco de código abierto que se utiliza para definir la infraestructura de aplicaciones sin servidor AWS, con algunos componentes adicionales que facilitan el trabajo con ellos. En este sentido, AWS SAM las plantillas son una extensión de CloudFormation las plantillas.

A continuación se muestra un ejemplo de una aplicación sin servidor básica. Esta aplicación procesa las solicitudes para obtener todos los elementos de una base de datos mediante una solicitud HTTP. Consta de los elementos siguientes:

1. Función que contiene la lógica necesaria para procesar la solicitud.
2. Una API HTTP que sirve de comunicación entre el cliente (solicitante) y la aplicación.
3. Una base de datos para almacenar elementos.
4. Permisos para que la aplicación se ejecuta de forma segura.



El código de infraestructura de esta aplicación se puede definir en la siguiente plantilla de AWS SAM :

```
AWSTemplateFormatVersion: 2010-09-09
Transform: AWS::Serverless-2016-10-31
Resources:
  getAllItemsFunction:
    Type: AWS::Serverless::Function
    Properties:
      Handler: src/get-all-items.getAllItemsHandler
      Runtime: nodejs20.x
      Events:
        Api:
          Type: HttpApi
          Properties:
            Path: /
            Method: GET
      Connectors:
        MyConn:
          Properties:
```

```
    Destination:
      Id: SampleTable
      Permissions:
        - Read
SampleTable:
  Type: AWS::Serverless::SimpleTable
```

En 23 líneas de código, se define la siguiente infraestructura:

- Una función que utiliza el AWS Lambda servicio.
- Una API HTTP que utiliza el servicio Amazon API Gateway.
- Una base de datos que utiliza el servicio Amazon DynamoDB.
- Los permisos AWS Identity and Access Management (IAM) necesarios para que estos servicios interactúen entre sí.

Para aprovisionar esta infraestructura, la plantilla se implementa en CloudFormation. Durante la implementación, AWS SAM transforma las 23 líneas de código en la CloudFormation sintaxis necesaria para generar estos recursos. AWS La CloudFormation plantilla transformada contiene más de 200 líneas de código!

¿Transformada CloudFormation plantilla

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Resources": {
    "getAllItemsFunction": {
      "Type": "AWS::Lambda::Function",
      "Metadata": {
        "SamResourceId": "getAllItemsFunction"
      },
      "Properties": {
        "Code": {
          "S3Bucket": "amzn-s3-demo-source-bucket-1a4x26zbcdkqr",
          "S3Key": "what-is-app/a6f856abf1b2c4f7488c09b367540b5b"
        },
        "Handler": "src/get-all-items.getAllItemsHandler",
        "Role": {
          "Fn::GetAtt": [
            "getAllItemsFunctionRole",
            "Arn"
          ]
        }
      }
    }
  }
}
```

```
    },
    "Runtime": "nodejs12.x",
    "Tags": [
      {
        "Key": "lambda:createdBy",
        "Value": "SAM"
      }
    ]
  },
  "getAllItemsFunctionRole": {
    "Type": "AWS::IAM::Role",
    "Properties": {
      "AssumeRolePolicyDocument": {
        "Version": "2012-10-17",
        "Statement": [
          {
            "Action": [
              "sts:AssumeRole"
            ],
            "Effect": "Allow",
            "Principal": {
              "Service": [
                "lambda.amazonaws.com"
              ]
            }
          }
        ]
      }
    ],
    "ManagedPolicyArns": [
      "arn:aws:iam::aws:policy/service-role/AWSLambdaBasicExecutionRole"
    ],
    "Tags": [
      {
        "Key": "lambda:createdBy",
        "Value": "SAM"
      }
    ]
  },
  "getAllItemsFunctionApiPermission": {
    "Type": "AWS::Lambda::Permission",
    "Properties": {
      "Action": "lambda:InvokeFunction",
```

```

      "FunctionName": {
        "Ref": "getAllItemsFunction"
      },
      "Principal": "apigateway.amazonaws.com",
      "SourceArn": {
        "Fn::Sub": [
          "arn:${AWS::Partition}:execute-api:${AWS::Region}:${AWS::AccountId}:
${__ApiId__}/${__Stage__}/GET/",
          {
            "__ApiId__": {
              "Ref": "ServerlessHttpApi"
            },
            "__Stage__": "*"
          }
        ]
      }
    },
    "ServerlessHttpApi": {
      "Type": "AWS::ApiGatewayV2::Api",
      "Properties": {
        "Body": {
          "info": {
            "version": "1.0",
            "title": {
              "Ref": "AWS::StackName"
            }
          },
          "paths": {
            "/": {
              "get": {
                "x-amazon-apigateway-integration": {
                  "httpMethod": "POST",
                  "type": "aws_proxy",
                  "uri": {
                    "Fn::Sub": "arn:${AWS::Partition}:apigateway:
${AWS::Region}:lambda:path/2015-03-31/functions/${getAllItemsFunction.Arn}/invocations"
                  },
                  "payloadFormatVersion": "2.0"
                },
                "responses": {}
              }
            }
          }
        }
      }
    },
  },

```

```
    "openapi": "3.0.1",
    "tags": [
      {
        "name": "httpapi:createdBy",
        "x-amazon-apigateway-tag-value": "SAM"
      }
    ]
  }
},
"ServerlessHttpApiApiGatewayDefaultStage": {
  "Type": "AWS::ApiGatewayV2::Stage",
  "Properties": {
    "ApiId": {
      "Ref": "ServerlessHttpApi"
    },
    "StageName": "$default",
    "Tags": {
      "httpapi:createdBy": "SAM"
    },
    "AutoDeploy": true
  }
},
"SampleTable": {
  "Type": "AWS::DynamoDB::Table",
  "Metadata": {
    "SamResourceId": "SampleTable"
  },
  "Properties": {
    "AttributeDefinitions": [
      {
        "AttributeName": "id",
        "AttributeType": "S"
      }
    ],
    "KeySchema": [
      {
        "AttributeName": "id",
        "KeyType": "HASH"
      }
    ],
    "BillingMode": "PAY_PER_REQUEST"
  }
},
```



```

"getAllItemsFunctionMyConnPolicy": {
  "Type": "AWS::IAM::ManagedPolicy",
  "Metadata": {
    "aws:sam:connectors": {
      "getAllItemsFunctionMyConn": {
        "Source": {
          "Type": "AWS::Serverless::Function"
        },
        "Destination": {
          "Type": "AWS::Serverless::SimpleTable"
        }
      }
    }
  },
  "Properties": {
    "PolicyDocument": {
      "Version": "2012-10-17",
      "Statement": [
        {
          "Effect": "Allow",
          "Action": [
            "dynamodb:GetItem",
            "dynamodb:Query",
            "dynamodb:Scan",
            "dynamodb:BatchGetItem",
            "dynamodb:ConditionCheckItem",
            "dynamodb: PartiQLSelect"
          ],
          "Resource": [
            {
              "Fn::GetAtt": [
                "SampleTable",
                "Arn"
              ]
            },
            {
              "Fn::Sub": [
                "${DestinationArn}/index/*",
                {
                  "DestinationArn": {
                    "Fn::GetAtt": [
                      "SampleTable",
                      "Arn"
                    ]
                  }
                ]
              ]
            }
          ]
        }
      ]
    }
  }
}

```

```
    }
  }
]
}
]
},
"Roles": [
  {
    "Ref": "getAllItemsFunctionRole"
  }
]
}
}
```

Al AWS SAM utilizarla, se definen 23 líneas de código de infraestructura. AWS SAM transforma el código en las más de 200 líneas de CloudFormation código necesarias para aprovisionar la aplicación.

¿Qué es el AWS SAM CLI?

AWS SAMCLI es una herramienta de línea de comandos que puede usar con AWS SAM plantillas e integraciones de terceros compatibles para crear y ejecutar sus aplicaciones sin servidor. Use la CLI de AWS SAM para:

- Inicializar rápidamente un nuevo proyecto de aplicación.
- Crear tu aplicación para su implementación.
- Realizar pruebas y depuraciones locales.
- Implementa la aplicación.
- Configure las canalizaciones CI/CD de implementación.
- Supervisar y solucionar los problemas de tu aplicación en la nube.
- Sincronizar los cambios locales con la nube a medida que ve desarrollándose.
- Y mucho más.

Se AWS SAMCLI utiliza mejor cuando se utiliza con CloudFormation plantillas AWS SAM y. También funciona con productos de terceros, como Terraform.

Inicio de un nuevo proyecto

Selecciona una de las plantillas iniciales o elige una ubicación de plantilla personalizada para comenzar un nuevo proyecto.

A continuación se muestra el comando `aws-sam init` para iniciar un nuevo proyecto de aplicación. Para empezar, seleccionamos el proyecto Hello World Example. La CLI de AWS SAM descarga una plantilla inicial y crea la estructura de directorios de carpetas del proyecto.

```
→ what-is sam init

You can preselect a particular runtime or package type when using the `sam init` experience.
Call `sam init --help` to learn more.

Which template source would you like to use?
  1 - AWS Quick Start Templates
  2 - Custom Template Location
Choice: 1

Choose an AWS Quick Start application template
  1 - Hello World Example
  2 - Multi-step workflow
  3 - Serverless API
  4 - Scheduled task
  5 - Standalone function
  6 - Data processing
  7 - Infrastructure event management
  8 - Serverless Connector Hello World Example
  9 - Multi-step workflow with Connectors
 10 - Lambda EFS example
 11 - Machine Learning
Template: 1

Use the most popular runtime and package type? (Python and zip) [y/N]:
```

Para obtener más información, consulta [Crea tu aplicación en AWS SAM](#).

Compilación de tu aplicación para implementación

Empaquete las dependencias de sus funciones y organice el código del proyecto y la estructura de carpetas para prepararlos para la implementación.

En este caso, utilizamos el comando `aws-sam build` para preparar nuestra aplicación para su implementación. La CLI de AWS SAM crea un directorio `.aws-sam` y organiza allí las dependencias y los archivos de nuestras aplicaciones para su implementación.

```
→ sam-app sam build
Building codeuri: /Users/evzz/Demo/what-is/sam-app/hello_world runtime: python3.9 metadata: {} architecture: x86_64 functions: HelloWorldFunction
Running PythonPipBuilder:ResolveDependencies
Running PythonPipBuilder:CopySource

Build Succeeded

Built Artifacts  : .aws-sam/build
Built Template   : .aws-sam/build/template.yaml

Commands you can use next
=====
[*] Validate SAM template: sam validate
[*] Invoke Function: sam local invoke
[*] Test Function in the Cloud: sam sync --stack-name {{stack-name}} --watch
[*] Deploy: sam deploy --guided
→ sam-app cd .aws-sam
→ .aws-sam ls
build          build.toml
→ .aws-sam
```

Para obtener más información, consulta [Creación de su aplicación](#).

Realización de testeo y depuraciones locales

En la máquina local, simule eventos, pruebe las API, invoque funciones y mucho más para depurar y probar tu aplicación.

En este caso, utilizamos el comando `sam local invoke` para invocar `HelloWorldFunction` de forma local. Para ello, la CLI de AWS SAM crea un contenedor local, crea nuestra función, la invoca y genera los resultados. Puede utilizar una aplicación como Docker para ejecutar contenedores en su máquina.

```
→ sam-app sam local invoke HelloWorldFunction
Invoking app.lambda_handler (python3.9)
Local image was not found.
Removing rapid images for repo public.ecr.aws/sam/emulation-python3.9
Building image.....
.....
Using local image: public.ecr.aws/lambda/python:3.9-rapid-x86_64.

Mounting /Users/evzz/Demo/what-is/sam-app/.aws-sam/build/HelloWorldFunction as /var/task:ro,delegated
inside runtime container
START RequestId: 6f8347ce-6b04-4246-a0de-6dc37f0eef51 Version: $LATEST
END RequestId: 6f8347ce-6b04-4246-a0de-6dc37f0eef51
REPORT RequestId: 6f8347ce-6b04-4246-a0de-6dc37f0eef51 Init Duration: 1.23 ms Duration: 639.26 ms B
illed Duration: 640 ms Memory Size: 128 MB Max Memory Used: 128 MB
{"statusCode": 200, "body": "{\"message\": \"hello world\"}"}
```

Para obtener más información, consulta [Prueba de la aplicación](#) y [Depuración de la aplicación](#).

Implementar la aplicación

Configure los ajustes de despliegue de su aplicación e impleméntela AWS en la nube para aprovisionar sus recursos.

Aquí, utilizamos el comando `sam deploy --guided` para implementar nuestra aplicación a través de un flujo interactivo. Esto nos AWS SAMCLI guía a través de la configuración de los ajustes de despliegue de nuestra aplicación CloudFormation, transforma nuestra plantilla en ella y la implementa CloudFormation para crear nuestros recursos.

```
→ sam-app sam deploy --guided

Configuring SAM deploy
=====

Looking for config file [samconfig.toml] : Not found

Setting default arguments for 'sam deploy'
=====
Stack Name [sam-app]:
AWS Region [us-west-2]:
#Shows you resources changes to be deployed and require a 'Y' to initiate deploy
Confirm changes before deploy [y/N]:
#SAM needs permission to be able to create roles to connect to the resources in your template
Allow SAM CLI IAM role creation [Y/n]:
#Preserves the state of previously provisioned resources when an operation fails
Disable rollback [y/N]:
HelloWorldFunction may not have authorization defined, Is this okay? [y/N]: y
Save arguments to configuration file [Y/n]:
SAM configuration file [samconfig.toml]:
SAM configuration environment [default]:

Looking for resources needed for deployment:
Managed S3 bucket: aws-sam-cli-managed-default-samclisourcebucket-1a4x26zbcdkqr
A different default S3 bucket can be set in samconfig.toml
```

Para obtener más información, consulte [Implementación de la aplicación y los recursos](#).

Configure las canalizaciones CI/CD de implementación

Cree canalizaciones de integración y entrega continuas y seguras (CI/CD) mediante un sistema compatible CI/CD .

En este caso, utilizamos el `sam pipeline init --bootstrap` comando para configurar una canalización CI/CD de implementación para nuestra aplicación. Nos AWS SAMCLI guía a través de nuestras opciones y genera los AWS recursos y el archivo de configuración para usarlos con nuestro CI/CD sistema.

[3] Reference application build resources

Enter the pipeline execution role ARN if you have previously created one, or we will create one for you :

Enter the CloudFormation execution role ARN if you have previously created one, or we will create one for you :

Please enter the artifact bucket ARN for your Lambda function. If you do not have a bucket, we will create one for you :

Does your application contain any IMAGE type Lambda functions? [y/N]: n

[4] Summary

Below is the summary of the answers:

- 1 - Account: 513423067560
- 2 - Stage configuration name: dev
- 3 - Region: us-west-2
- 4 - Pipeline user: [to be created]
- 5 - Pipeline execution role: [to be created]
- 6 - CloudFormation execution role: [to be created]
- 7 - Artifacts bucket: [to be created]
- 8 - ECR image repository: [skipped]

Press enter to confirm the values above, or select an item to edit the value:

This will create the following required resources for the 'dev' configuration:

- Pipeline IAM user
- Pipeline execution role
- CloudFormation execution role
- Artifact bucket

Should we proceed with the creation? [y/N]: ☐

Para obtener más información, consulta [Implemente con CI/CD sistemas y canalizaciones](#).

Supervisión y solución de los problemas de la aplicación en la nube

Consulta información importante sobre los recursos desplegados, recopile registros y utiliza herramientas de supervisión integradas, como AWS X-Ray.

En este caso, utilizamos el comando `sam list` para ver los recursos desplegados. Obtenemos el punto de conexión de la API y lo invocamos, lo que activa nuestra función. Luego, usamos `sam logs` para ver los registros de la función.

```
→ sam-app sam logs --stack-name sam-app
2023/03/13/[$LATEST]0a433e844dd445bd82d0d78cd55e0cdc 2023-03-13T21:06:42.075000 INIT_START Runtime Version: python:3.9.v16 Runtime Version ARN: arn:aws:lambda:us-west-2::runtime:07a48df201798d627f2b950f03bb227aab4a655a1d019c3296406f95937e2525
2023/03/13/[$LATEST]0a433e844dd445bd82d0d78cd55e0cdc 2023-03-13T21:06:42.180000 START RequestId: 778e4226-0a80-435f-929b-5b19292ed9a7 Version: $LATEST
2023/03/13/[$LATEST]0a433e844dd445bd82d0d78cd55e0cdc 2023-03-13T21:06:42.181000 END RequestId: 778e4226-0a80-435f-929b-5b19292ed9a7
2023/03/13/[$LATEST]0a433e844dd445bd82d0d78cd55e0cdc 2023-03-13T21:06:42.182000 REPORT RequestId: 778e4226-0a80-435f-929b-5b19292ed9a7 Duration: 1.69 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memory Used: 36 MB Init Duration: 104.13 ms
```

Para obtener más información, consulta [Supervisión de la aplicación](#).

Sincronización de los cambios locales con la nube a medida que va desarrollando

A medida que va desarrollando en tu máquina local, sincronice automáticamente los cambios con la nube. Consulta rápidamente los cambios y realiza pruebas y validaciones en la nube.

En este caso, utilizaremos el comando `sam sync --watch` para que la CLI de AWS SAM vigile los cambios locales. Modificaremos el código `HelloWorldFunction` y, de forma automática, la CLI de AWS SAM detectará el cambio e implementará las actualizaciones en la nube.


```
-----
Key           HelloWorldFunctionIamRole
Description   Implicit IAM Role created for Hello World function
Value         arn:aws:iam::513423067560:role/sam-app-HelloWorldFunctionRole-15GL0UR9LMT1W

Key           HelloWorldApi
Description   API Gateway endpoint URL for Prod stage for Hello World function
Value         https://ets1gv8lxi.execute-api.us-west-2.amazonaws.com/Prod/hello/

Key           HelloWorldFunction
Description   Hello World Lambda Function ARN
Value         arn:aws:lambda:us-west-2:513423067560:function:sam-app-HelloWorldFunction-
yQDNe17r9maD
-----

Stack update succeeded. Sync infra completed.

Infra sync completed.
CodeTrigger not created as CodeUri or DefinitionUri is missing for ServerlessRestApi.
Syncing Lambda Function HelloWorldFunction...
Manifest is not changed for (HelloWorldFunction), running incremental build
Building codeuri: /Users/evzz/Demo/what-is/sam-app/hello_world runtime: python3.9 metadata: {} architecture: x86_64 functions: HelloWorldFunction
Running PythonPipBuilder:CopySource
Finished syncing Lambda Function HelloWorldFunction.
█
```

Prueba de los recursos compatibles en la nube

Invoca y transfiere eventos a los recursos compatibles en la nube.

En este caso, utilizamos el comando `sam remote invoke` para probar una función de Lambda implementada en la nube. Invocamos la función de Lambda y recibimos sus registros y su respuesta. Con la función de Lambda configurada para transmitir las respuestas, la CLI de AWS SAM transmite su respuesta en tiempo real.

```
→ lambda-streaming-app sam remote invoke StreamingFunction --stack-name lambda-streaming-app
```

Más información

Para seguir informándose AWS SAM, consulta los siguientes recursos:

- [El AWS SAM taller completo](#): un taller diseñado para enseñarle muchas de las principales funciones que AWS SAM ofrece.
- [Sesiones con SAM](#): serie de vídeos sobre su uso AWS SAM creada por nuestro equipo de AWS Serverless Developer Advocate.
- [Serverless Land](#): sitio que reúne la información más reciente, blogs, vídeos, códigos y recursos de aprendizaje sobre sistemas sin servidor. AWS

Siguientes pasos

Si es la primera vez que lo usa AWS SAM, consulte. [Empezar con AWS SAM](#)

Conceptos sin servidor para AWS Serverless Application Model

Obtenga información sobre los conceptos básicos de la tecnología sin servidor antes de usar AWS Serverless Application Model (AWS SAM).

Conceptos sobre sistemas sin servidor

Arquitectura basada en eventos

Una aplicación sin servidor consta de AWS servicios individuales, como los de computación y Amazon DynamoDB AWS Lambda para la administración de bases de datos, cada uno de los cuales desempeña una función especializada. A continuación, estos servicios se integran de forma flexible entre sí mediante una arquitectura basada en eventos. Para obtener más información sobre la arquitectura basada en eventos, consulta [¿Qué es una arquitectura basada en eventos?](#)

Infraestructura como código (IaC)

La infraestructura como código (IaC) es una forma de tratar la infraestructura de la misma manera que los desarrolladores tratan el código, aplicando el mismo rigor que el desarrollo del código de aplicaciones al aprovisionamiento de la infraestructura. Usted define su infraestructura en un archivo de plantilla, la implementa y AWS crea los recursos automáticamente. AWS Con IaC, usted define en el código lo que quiere AWS aprovisionar. Para ver una comparación de las herramientas de IaC disponibles AWS, consulte. [Infraestructura como código \(IaC\)](#)

Tecnologías sin servidor

Con las tecnologías AWS sin servidor, puede crear y ejecutar aplicaciones sin tener que administrar sus propios servidores. Toda la administración de los servidores se lleva a cabo de forma automática AWS, lo que proporciona numerosas ventajas, como el escalado automático y la alta disponibilidad integrada, lo que le permite llevar su idea a la producción rápidamente. Al utilizar tecnologías sin servidor, puede centrarse en el núcleo de su producto sin tener que preocuparse por la administración y el funcionamiento de los servidores. Para obtener más información sobre sin servidor, consulta lo siguiente:

- [Sin servidor activado AWS](#)
- [Guía para desarrolladores sin servidor](#): proporciona una visión general conceptual del desarrollo sin servidor en AWS Cloud.

Para obtener una introducción básica a los principales servicios AWS sin servidor, consulte [Serverless 101: Comprender los servicios sin servidor en Serverless Land](#).

Aplicación sin servidor

Cuando los usa AWS SAM, administra los recursos relacionados en una aplicación, que consiste en su proyecto y su AWS SAM plantilla. Todos los recursos de la aplicación están definidos o

se hace referencia a ellos en la AWS SAM plantilla. Cuando AWS SAM procesa la plantilla, crea CloudFormation recursos. En CloudFormation, los recursos se administran en una sola unidad denominada pila, y todos los recursos de una pila se definen mediante la CloudFormation plantilla de la pila.

Infraestructura como código (IaC)

Con la infraestructura como código (IaC), puede automatizar la implementación y la administración de sus AWS recursos, incluidas las aplicaciones sin servidor. La IaC permite definir la infraestructura mediante código, lo que facilita la creación de versiones, el uso compartido y la replicación de las implementaciones. Este enfoque ayuda a realizar lo siguiente:

- Acelerar el ciclo de desarrollo
- Facilitar la administración de la configuración
- Mejorar la fiabilidad y coherencia de las implementaciones

Herramientas de IaC para AWS aplicaciones sin servidor

AWS ofrece varias herramientas de IaC para ayudarlo a crear, implementar y administrar sus recursos en la nube. En esta sección se explica cómo AWS SAM se integra en este ecosistema y cómo funciona con otras herramientas de AWS IaC.

AWS CloudFormation

Con ellas [CloudFormation](#), puedes modelar y aprovisionar toda tu AWS infraestructura con plantillas YAML o JSON. CloudFormation gestiona la creación, actualización y eliminación de recursos de forma automática. Al implementar AWS SAM aplicaciones, CloudFormation procesa las plantillas transformadas para crear y administrar sus recursos.

AWS Serverless Application Model (AWS SAM)

AWS SAM le ayuda a crear aplicaciones sin servidor con una sintaxis simplificada para definir los recursos sin servidor. Puede usar AWS SAM plantillas para aprovisionar funciones de eventos de Lambda, API, bases de datos y fuentes de eventos mediante una sintaxis YAML concisa. AWS SAM transforma estas plantillas en CloudFormation plantillas durante la implementación.

Note

Si bien AWS SAM se especializa en aplicaciones sin servidor, puede usar cualquier tipo CloudFormation de recurso en sus AWS SAM plantillas. Esto confiere flexibilidad para incluir recursos sin servidor cuando sea necesario.

AWS Kit Cloud Development (AWS CDK)

Con [AWS CDK](#), puede definir su infraestructura sin servidor utilizando lenguajes de programación conocidos como Python TypeScript, Java, C#/.Net o Go. Puede utilizar estructuras de programación, como bucles y condiciones, para definir su infraestructura y generar plantillas para su implementación. AWS CDK CloudFormation Puede utilizarlas para probar y AWS SAMCLI depurar localmente las aplicaciones creadas con ellas. AWS CDK Para obtener más información, consulte [Probar aplicaciones de CDK de manera local](#).

Comparación de herramientas de IaC para aplicaciones sin servidor

Para elegir una herramienta de IaC para sus aplicaciones sin servidor, es importante tener presente las preferencias de su equipo, los requisitos del proyecto y los flujos de trabajo existentes. En la siguiente tabla se comparan las características clave de las herramientas de AWS IaC para el desarrollo sin servidor:

Herramienta	Uso principal	Lo mejor para	Funciona con AWS SAM	Cuándo elegir
CloudFormation	Administrar una AWS infraestructura compleja	Aplicaciones que requieren un control detallado de AWS los recursos	AWS SAM las plantillas se transforman en CloudFormation plantillas durante el despliegue	Para un control detallado sobre recursos que no están sin servidor
AWS SAM	Desarrollo de aplicaciones sin servidor	Equipos que crean aplicaciones sin servidor con Lambda	Funcionalidad nativa	Cuando se centra principalmente en arquitecturas sin servidor con

Herramienta	Uso principal	Lo mejor para	Funciona con AWS SAM	Cuándo elegir
				funciones de Lambda, API Gateway, API y otros recursos sin servidor
AWS CDK	Definición de infraestructura mediante lenguajes de programación	Equipos que prefieran los lenguajes tipados y el enfoque basado en el código	Genere AWS SAM plantillas y utilícelas AWS SAMCLI para realizar pruebas	Cuando necesite una definición de infraestructura programática o una lógica de configuración de recursos compleja

Note

Si bien esta guía se centra en las herramientas AWS de iAC nativas, Terraform es otra solución popular de iAC que se puede utilizar para definir aplicaciones sin servidor. La CLI de AWS SAM admite las pruebas locales de las funciones de Lambda definidas en Terraform. Para obtener más información, consulte [AWS SAM CLI Terraform compatibilidad](#).

Más información

- [Para obtener más información sobre DevOps las prácticas sobre, consulte la Introducción a on AWS DevOps AWS](#)
- Para obtener información sobre el uso de Lambda con diferentes herramientas de IaC, consulte [Uso de Lambda con la infraestructura como código \(IaC\)](#)

Empezar con AWS SAM

Comience AWS SAM revisando y completando los temas de esta sección. [AWS SAM requisitos previos](#) proporciona instrucciones detalladas sobre la configuración de una AWS cuenta, la creación de usuarios de IAM, la creación de claves de acceso y la AWS SAMCLI instalación y configuración de. Después de completar los requisitos previos, cree su enfoque de desarrollo:

- [Instala la AWS SAM CLI](#), lo que puede hacer en los sistemas operativos Linux, Windows y macOS.
- [Cómo empezar con AWS SAM con VS Code Toolkit](#) para disfrutar de una experiencia de desarrollo visual con funciones integradas de depuración y pruebas.

Una vez completada la instalación, si lo desea, puede seguir el tutorial de AWS SAM Hello World. Al seguir este tutorial, se detalla el proceso de creación de una aplicación básica sin servidor con AWS SAM. Tras completar el tutorial, tendrá todo listo para revisar los conceptos que se detallan en [Cómo usar AWS Serverless Application Model \(AWS SAM\)](#).

Temas

- [AWS SAM requisitos previos](#)
- [Instala la AWS SAM CLI](#)
- [Tutorial: Implemente una aplicación Hello World con AWS SAM](#)
- [Conversión de funciones de eventos de Lambda en AWS SAM Aplicaciones](#)

AWS SAM requisitos previos

Complete los siguientes requisitos previos antes de instalar y utilizar la interfaz de línea de AWS Serverless Application Model comandos ()AWS SAMCLI.

Para usar el AWS SAMCLI, necesita lo siguiente:

- Una AWS cuenta, credenciales AWS Identity and Access Management (IAM) y un key pair de claves de acceso de IAM.
- El AWS Command Line Interface (AWS CLI) para configurar AWS las credenciales.

Temas

- [Inscríbase en una Cuenta de AWS](#)

- [Paso 1: Instale el AWS CLI](#)
- [Paso 2: utilice el AWS CLI para configurar AWS credentials](#)
- [Paso 3: Instalar \(opcional\) AWS Kit de herramientas para VS Code](#)
- [Siguiendo pasos](#)

Inscríbase en una Cuenta de AWS

Para empezar AWS, necesitas un Cuenta de AWS. Para obtener información sobre cómo crear un Cuenta de AWS, consulte [Cómo empezar con un Cuenta de AWS](#) en la Guía de AWS Account Management referencia.

Paso 1: Instale el AWS CLI

AWS CLI Se trata de una herramienta de código abierto que permite interactuar con ella Servicios de AWS mediante comandos de la consola de la línea de comandos. Lo AWS SAMCLI requiere AWS CLI para actividades como la configuración de credenciales. Para obtener más información sobre el AWS CLI, consulte [¿Qué es el AWS Command Line Interface?](#) en la Guía AWS Command Line Interface del usuario.

Para instalarlo AWS CLI, consulte [Instalación o actualización de la última versión de AWS CLI en la Guía del AWS Command Line Interface usuario](#).

Paso 2: utilice el AWS CLI para configurar AWS credentials

Cómo configurar las credenciales con IAM Identity Center

- Para configurar las credenciales con el Centro de identidades de IAM, consulte [Configurar su perfil con el asistente de AWS configuración de sso](#).

Para configurar las credenciales con AWS CLI

1. Ejecuta el comando `aws configure` desde la línea de comandos.
2. Configura lo siguiente. Seleccione cada enlace para obtener más información:
 - a. [ID de clave de acceso](#)
 - b. [Clave de acceso secreta](#)
 - c. [Región de AWS](#)

d. [Formato de salida](#)

En el ejemplo siguiente se muestran los valores de ejemplo.

```
$ aws configure
AWS Access Key ID [None]: AKIAIOSFODNN7EXAMPLE
AWS Secret Access Key [None]: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
Default region name [None]: us-west-2
Default output format [None]: json
```

AWS CLI Almacena esta información en un perfil (un conjunto de ajustes) denominado `default` en los `config` archivos `credentials` and. Estos archivos están ubicados en el archivo `.aws` en su directorio personal. De forma predeterminada, la información de este perfil se utiliza cuando se ejecuta un AWS CLI comando que no especifica de forma explícita el perfil que se va a utilizar. Para obtener más información sobre el archivo `credentials`, consulta [Ajustes del archivo de configuración y credenciales](#) en la Guía del usuario de AWS Command Line Interface .

Para obtener más información sobre la configuración de credenciales, como el uso de un archivo de configuración y credenciales existente, consulta [Quick setup](#) en la Guía del usuario de AWS Command Line Interface .

Paso 3: Instalar (opcional) AWS Kit de herramientas para VS Code

Para los desarrolladores que prefieren un entorno de desarrollo integrado, el AWS Toolkit for VS Code ofrece potentes funciones para el desarrollo sin servidor, como la depuración `visualCodeLens` , la integración y los flujos de trabajo de implementación simplificados.

Requisitos previos para el desarrollo de VS Code

- Visual Studio Code (versión 1.73.0 o una versión posterior) está instalado en el sistema
- Extensión de compatibilidad con el lenguaje YAML para VS Code

Para instalar la de AWS Kit de herramientas para VS Code

1. Abra Visual Studio Code
2. Abra a la vista de extensiones (Ctrl+Shift+X o Cmd+Shift+X)
3. Busque "Toolkit»AWS

4. Instale la extensión «AWS Toolkit» de Amazon Web Services
5. Instale la extensión «YAML» de Red Hat (necesaria para las funciones de la plantilla CodeLens SAM)

Ventajas de usar VS Code con SAM AWS

- Depuración visual: establezca puntos de interrupción y revise sus funciones de eventos de Lambda de forma local
- CodeLens integración: cree, implemente e invoque funciones directamente desde su plantilla SAM
- Terminal integrado: acceda a AWS CLI los comandos de AWS SAM sin salir del editor
- Validación de plantillas: Real-time validación y IntelliSense para plantillas SAM

Para obtener información sobre cómo configurar sus AWS credenciales en VS Code, consulte [Configuración de credenciales](#) en la Guía del usuario de AWS Toolkit for VS Code.

Siguientes pasos

- Ahora está listo para instalar AWS SAMCLI y empezar a usarlo AWS SAM. Para instalar el AWS SAMCLI, consulte [Instala la AWS SAM CLI](#).
- Para configurar Visual Studio Code para el desarrollo sin servidor, consulte [Configuración del AWS Toolkit for VS Code](#).

Instala la AWS SAM CLI

Instale la versión más reciente de la interfaz de línea de AWS Serverless Application Model comandos (AWS SAMCLI) en los sistemas operativos compatibles siguiendo las instrucciones de [Paso 1: Instale el AWS CLI](#).

Para obtener información sobre la administración de una versión instalada actualmente de AWS SAMCLI, incluido cómo actualizar, desinstalar o administrar las compilaciones nocturnas, consulte [Administración de versiones de AWS SAM CLI](#).

 ¿Es la primera vez que instala AWS SAM CLI?

Completa todos los [requisitos previos](#) de la sección anterior antes de continuar. Esto incluye:

1. Registrarse para obtener una AWS cuenta.

2. Configurar el acceso seguro a AWS.
3. Creación de un identificador de claves de acceso y de una claves de acceso secreta.
4. Instalación del AWS CLI.
5. Configuración de AWS credenciales.

Temas

- [Instalación de la AWS SAM CLI](#)
- [Solución de errores de la instalación](#)
- [Siguiendo pasos](#)
- [Opcional: compruebe la integridad del AWS SAM CLI Instalador de](#)

Instalación de la AWS SAM CLI

Note

A partir de septiembre de 2023, AWS dejará de mantener el Homebrew instalador AWS gestionado de AWS SAMCLI (`aws/tap/aws-sam-cli`). Si lo utiliza Homebrew para instalar y administrar AWS SAMCLI, consulte las siguientes opciones:

- Para seguir utilizando Homebrew, puedes utilizar el instalador gestionado por la comunidad. Para obtener más información, consulta [Administración de CLI de AWS SAM con la Homebrew](#).
- Se recomienda utilizar uno de los métodos de instalación propios que se documentan en esta página. Antes de utilizar uno de estos métodos, consulta [Cambiar de Homebrew](#).
- Para obtener más información, consulte [Release version: 1.121.0](#).

Para instalarlo AWS SAMCLI, siga las instrucciones de su sistema operativo.

Linux

x86_64 - command line installer

1. Descarga el [archivo .zip de AWS SAM CLI](#) a un directorio de tu elección.

2. (Opcional) Puedes comprobar la integridad del instalador antes de la instalación. Para obtener instrucciones, consulta [Opcional: compruebe la integridad del AWS SAM CLI Instalador de](#).
3. Descomprime los archivos de instalación en el directorio que prefieras. A continuación, se muestra un ejemplo de uso del subdirectorio `sam-installation`.

 Note

Si el sistema operativo no tiene el comando `unzip` integrado, utiliza un equivalente.

```
$ unzip aws-sam-cli-linux-x86_64.zip -d sam-installation
```

4. Instala la AWS SAM CLI ejecutando el `install` ejecutable. Este ejecutable se ubica en el directorio utilizado en el paso anterior. A continuación, se muestra un ejemplo de uso del subdirectorio `sam-installation`:

```
$ sudo ./sam-installation/install
```

5. Verifica la instalación.

```
$ sam --version
```

Para confirmar que la instalación se ha realizado correctamente, debería aparecer un resultado que sustituya el siguiente texto entre corchetes por la última versión disponible:

```
SAM CLI, <latest version>
```

arm64 - command line installer

1. Descarga el [archivo .zip de AWS SAM CLI](#) a un directorio de tu elección.
2. (Opcional) Puedes comprobar la integridad del instalador antes de la instalación. Para obtener instrucciones, consulta [Opcional: compruebe la integridad del AWS SAM CLI Instalador de](#).
3. Descomprime los archivos de instalación en el directorio que prefieras. A continuación, se muestra un ejemplo de uso del subdirectorio `sam-installation`.

 Note

Si el sistema operativo no tiene el comando unzip integrado, utiliza un equivalente.

```
$ unzip aws-sam-cli-linux-arm64.zip -d sam-installation
```

4. Instala la AWS SAM CLI ejecutando el `install` ejecutable. Este ejecutable se ubica en el directorio utilizado en el paso anterior. A continuación, se muestra un ejemplo de uso del subdirectorio `sam-installation`:

```
$ sudo ./sam-installation/install
```

5. Verifica la instalación.

```
$ sam --version
```

Para confirmar que la instalación se ha realizado correctamente, debería aparecer un resultado como el siguiente, pero que reemplace el texto entre corchetes por la última versión de la CLI de SAM:

```
SAM CLI, <latest version>
```

macOS

Pasos de instalación

Utilice el instalador del paquete para instalar el AWS SAMCLI. Además, el instalador del paquete tiene dos métodos de instalación entre los que puede elegir: GUI y Línea de comandos. Puede instalarlo para todos los usuarios o solo para el usuario actual. Para realizar la instalación para todos los usuarios, se requiere la autorización del superusuario.

 Note

AWS SAM La CLI no admite versiones de macOS anteriores a macOS 13.x. Actualice a una versión compatible de macOS o instale AWS SAM CLI con Homebrew.

GUI - All users

Descarga del instalador del paquete para instalar la CLI de AWS SAM

 Note

Si instalaste previamente la AWS SAM CLI mediante Homebrew o pip, primero debes desinstalarlo. Para obtener instrucciones, consulta [Desinstalación de la AWS SAM CLI](#).

1. Descargue el pkg para macOS al directorio que elija:

- Para equipos Mac con procesadores Intel, elija x86_64 — [-x86_64.pkg aws-sam-cli-macos](#)
- [Para equipos Mac con Apple Silicon, selecciona arm64 — -arm64.pkg aws-sam-cli-macos](#)

 Note

Tiene la opción de comprobar la integridad del instalador antes de la instalación. Para obtener instrucciones, consulta [Opcional: compruebe la integridad del AWS SAM CLI Instalador de](#).

2. Ejecuta el archivo descargado y sigue las instrucciones que aparecen en pantalla para continuar con los pasos de introducción, léeme y licencia.
3. En Destination Select, selecciona Instalar para todos los usuarios de este equipo.
4. Para el tipo de instalación, elige dónde se instalará la CLI de AWS SAM y pulsa Instalar. La ubicación predeterminada recomendada es `/usr/local/aws-sam-cli`.

 Note

Para invocar la AWS SAM CLI con el comando `sam`, el instalador crea automáticamente un enlace simbólico entre `/usr/local/bin/sam` y `/usr/local/aws-sam-cli/sam` o la carpeta de instalación que elijas.

5. El AWS SAM CLI, procederán a la instalación y mostrarán el siguiente mensaje: «La instalación se ha realizado correctamente». Pulsa Cerrar.

Para comprobar que la instalación se ha realizado correctamente

- Comprueba que la AWS SAM CLI se ha instalado correctamente y que el enlace simbólico está configurado ejecutando:

```
$ which sam
/usr/local/bin/sam
$ sam --version
SAM CLI, <latest version>
```

GUI - Current user

Descarga e instalación de la CLI de AWS SAM

Note

Si instalaste previamente la AWS SAM CLI mediante Homebrew o pip, primero debes desinstalarlo. Para obtener instrucciones, consulta [Desinstalación de la AWS SAM CLI](#).

1. Descargue el pkg para macOS al directorio que elija:

- Para equipos Mac con procesadores Intel, elige [x86_64 — -x86_64.pkg aws-sam-cli-macos](#)
- [Para equipos Mac con Apple Silicon, selecciona arm64 — -arm64.pkg aws-sam-cli-macos](#)

Note

Tiene la opción de comprobar la integridad del instalador antes de la instalación. Para obtener instrucciones, consulta [Opcional: compruebe la integridad del AWS SAM CLI Instalador de](#).

2. Ejecuta el archivo descargado y sigue las instrucciones que aparecen en pantalla para continuar con los pasos de introducción, léeme y licencia.
3. En Destination Select, selecciona Instalar solo para mí. Si no ves esta opción, ve al paso siguiente.
4. En Tipos de instancias, haga lo siguiente:

1. Elige dónde se instalará la CLI de AWS SAM. La ubicación predeterminada es `/usr/local/aws-sam-cli`. Selecciona una ubicación para la que tengas permisos de escritura. Para cambiar la ubicación de instalación, selecciona local y elige su ubicación. Pulsa Continuar cuando hayas terminado.
2. Si en el paso anterior no tenías la opción de elegir Instalar solo para mí, selecciona Cambiar ubicación de instalación > Instalar solo para mí y pulsa Continuar.
3. Pulsa Instalar.
5. El AWS SAM CLI procederán a la instalación y mostrarán «La instalación se ha realizado correctamente». Pulsa Cerrar.

Para crear un enlace simbólico

- Para invocar AWS SAM CLI con el comando `sam`, debes crear manualmente un enlace simbólico entre la CLI de AWS SAM y su `$PATH`. Crea tu enlace simbólico modificando y ejecutando el siguiente comando:

```
$ sudo ln -s /path-to/aws-sam-cli/sam /path-to-symlink-directory/sam
```

- **sudo**— Si su usuario tiene permisos de escritura para `$PATH`, no `sudo` es obligatorio. De lo contrario, se requiere el `sudo`.
- **path-to**— Ruta a la que instalaste el AWS SAMCLI programa. Por ejemplo, `/Users/myUser/Desktop`.
- **path-to-symlink-directory**— Tu variable de `$PATH` entorno. La ubicación predeterminada es `/usr/local/bin`.

Para comprobar que la instalación se ha realizado correctamente

- Comprueba que la AWS SAM CLI se ha instalado correctamente y que el enlace simbólico está configurado ejecutando:

```
$ which sam  
/usr/local/bin/sam  
$ sam --version  
SAM CLI, <latest version>
```


Command line - All users

Descarga e instalación de la CLI de AWS SAM

Note

Si instalaste previamente la AWS SAM CLI mediante Homebrew o pip, primero debes desinstalarlo. Para obtener instrucciones, consulta [Desinstalación de la AWS SAM CLI](#).

1. Descargue el pkg para macOS al directorio que elija:

- Para ordenadores Mac con procesadores Intel, elija x86_64 — [-x86_64.pkg aws-sam-cli-macos](#)
- [Para equipos Mac con Apple Silicon, selecciona arm64 — -arm64.pkg aws-sam-cli-macos](#)

Note

Tiene la opción de comprobar la integridad del instalador antes de la instalación. Para obtener instrucciones, consulta [Opcional: compruebe la integridad del AWS SAM CLI Instalador de](#).

2. Modifica y ejecuta el script de instalación:

```
$ sudo installer -pkg path-to-pkg-installer/name-of-pkg-installer -target /  
installer: Package name is AWS SAM CLI  
installer: Upgrading at base path /  
installer: The upgrade was successful.
```

Note

Para invocar la CLI de AWS SAM con el sam comando, el instalador crea automáticamente un enlace simbólico entre `/usr/local/bin/sam` y `/usr/local/aws-sam-cli/sam`.

Para comprobar que la instalación se ha realizado correctamente

- Comprueba que la AWS SAM CLI se ha instalado correctamente y que el enlace simbólico está configurado ejecutando:

```
$ which sam
/usr/local/bin/sam
$ sam --version
SAM CLI, <latest version>
```

Command line - Current user

Descarga e instalación de la CLI de AWS SAM

 Note

Si instalaste previamente la AWS SAM CLI mediante Homebrew o pip, primero debes desinstalarlo. Para obtener instrucciones, consulta [Desinstalación de la AWS SAM CLI](#).

1. Descargue el pkg para macOS al directorio que elija:

- Para equipos Mac con procesadores Intel, elige [x86_64 — -x86_64.pkg aws-sam-cli-macos](#)
- [Para equipos Mac con Apple Silicon, selecciona arm64 — -arm64.pkg aws-sam-cli-macos](#)

 Note

Tiene la opción de comprobar la integridad del instalador antes de la instalación. Para obtener instrucciones, consulta [Opcional: compruebe la integridad del AWS SAM CLI Instalador de](#).

2. Define un directorio de instalación en el que tengas permisos de escritura. A continuación, cree un archivo xml con la plantilla y modifíquelo para que refleje el directorio de instalación. El directorio debe existir previamente.

Por ejemplo, si lo *path-to-my-directory* reemplaza por `/Users/myUser/Desktop`, la carpeta del `aws-sam-cli` programa se instalará allí.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
  <array>
    <dict>
      <key>choiceAttribute</key>
      <string>customLocation</string>
      <key>attributeSetting</key>
      <string>path-to-my-directory</string>
      <key>choiceIdentifier</key>
      <string>default</string>
    </dict>
  </array>
</plist>
```

3. Guarda el archivo xml y comprueba que es válido ejecutando lo siguiente:

```
$ installer -pkg path-to-pkg-installer \
-target CurrentUserHomeDirectory \
-showChoicesAfterApplyingChangesXML path-to-your-xml-file
```

El resultado debes mostrar las preferencias que se aplicarán al programa AWS SAM CLI.

4. Ejecute lo siguiente para instalar AWS SAMCLI:

```
$ installer -pkg path-to-pkg-installer \
-target CurrentUserHomeDirectory \
-applyChoiceChangesXML path-to-your-xml-file

# Example output
installer: Package name is AWS SAM CLI
installer: choices changes file 'path-to-your-xml-file' applied
installer: Upgrading at base path base-path-of-xml-file
installer: The upgrade was successful.
```

Para crear un enlace simbólico

- Para invocar AWS SAM CLI con el comando `sam`, debes crear manualmente un enlace simbólico entre la CLI de AWS SAM y su `$PATH`. Crea tu enlace simbólico modificando y ejecutando el siguiente comando:

```
$ sudo ln -s /path-to/aws-sam-cli/sam /path-to-symlink-directory/sam
```

- **`sudo`**— Si su usuario tiene permisos de escritura para `$PATH`, no `sudo` es obligatorio. De lo contrario, se requiere el `sudo`.
- **`path-to`**— Ruta a la que instalaste el AWS SAMCLI programa. Por ejemplo, `/Users/myUser/Desktop`.
- **`path-to-symlink-directory`**— Tu variable de `$PATH` entorno. La ubicación predeterminada es `/usr/local/bin`.

Para comprobar que la instalación se ha realizado correctamente

- Comprueba que la AWS SAM CLI se ha instalado correctamente y que el enlace simbólico está configurado ejecutando:

```
$ which sam  
/usr/local/bin/sam  
$ sam --version  
SAM CLI, <latest version>
```

Windows

Los archivos Windows Installer (MSI) son los archivos de instalación de paquetes para el sistema operativo Windows.

Siga estos pasos para instalar la CLI de AWS SAM mediante el archivo MSI.

1. Descarga la CLI de AWS SAM de [64 bits](#).
2. (Opcional) Puedes comprobar la integridad del instalador antes de la instalación. Para obtener instrucciones, consulta [Opcional: compruebe la integridad del AWS SAM CLI Instalador de](#).
3. Verifica la instalación.

Después de completar la instalación, verifícala abriendo una nueva línea de comandos o línea de PowerShell comandos. Debería poder invocar `sam` en la línea de comandos.

```
sam --version
```

Tras una instalación correcta del AWS SAMCLI, debería ver un resultado como el siguiente:

```
SAM CLI, <latest version>
```

4. Habilite las rutas largas (solo en Windows 10 y versiones posteriores).

Important

AWS SAMCLIEs posible que interactúen con las rutas de archivo que superen el límite máximo de rutas de Windows. Esto puede provocar errores durante la ejecución de `sam init` debido a las limitaciones de `MAX_PATH` en Windows 10. Para resolver este problema, debes configurar el comportamiento de las nuevas rutas de acceso largo.

Para habilitar las rutas largas, consulta [Habilitar las rutas largas en Windows 10, versión 1607 y versiones posteriores](#), en la documentación de desarrollo de aplicaciones de Microsoft Windows.

5. Instala Git

Para descargar aplicaciones de muestra mediante el comando `sam init`, también debes instalar Git. Para ver las instrucciones, consulta la [Instalación de Git](#).

Solución de errores de la instalación

Linux

Error de Docker: «No se puedes conectar al daemon de Docker. ¿Se está ejecutando el daemon de Docker en este host?»

En algunos casos, es posible que tengas que reiniciar la instancia para que `ec2-user` tengas los permisos necesarios para acceder al daemon de Docker. Si recibes este error, intenta reiniciar la instancia.

Error del intérprete de comandos: «no se ha encontrado el comando»

Si recibes este error, tu intérprete de comandos no puede localizar el ejecutable AWS SAMCLI en la ruta. Comprueba la ubicación del directorio en el que instalaste el ejecutable AWS SAMCLI y, a continuación, comprueba que el directorio se encuentra en su ruta.

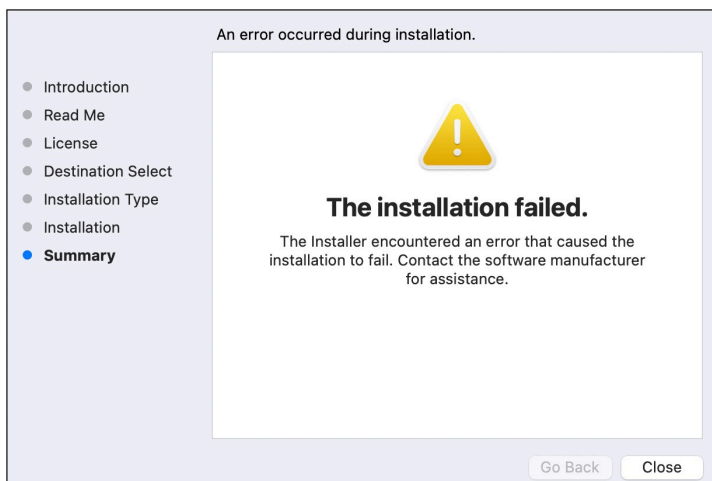
AWS SAMCLIerror: «/lib64/libc.so.6: no se ha encontrado la versión `GLIBC_2.14' (requerida por ./so.1)» usr/local/aws-sam-cli/dist/libz

Si recibes este error, significa que estás usando una versión no admitida de Linux y que la versión integrada de glibc es obsoleta. Haz una de estas dos operaciones:

- Actualiza el host de Linux a la versión de 64 bits de una distribución reciente de CentOS, Fedora, Ubuntu o Amazon Linux 2.
- Siga estas instrucciones para [Instala la AWS SAM CLI](#).

macOS

Error en la instalación



Si estás instalando la AWS SAMCLI para tu usuario y has seleccionado un directorio de instalación para el que no tienes permisos de escritura, podría producirse este error. Haz una de estas dos operaciones:

1. Selecciona otro directorio de instalación para el que tengas permisos de escritura.
2. Elimine el instalador. A continuación, descárgala y ejecútalo de nuevo.

Siguientes pasos

Para obtener más información sobre las aplicaciones sin servidor de AWS SAM CLI y comenzar a crear las suyas, consulta los siguientes enlaces:

- [Tutorial: Implemente una aplicación Hello World con AWS SAM](#)— Step-by-step instrucciones para descargar, crear e implementar una aplicación básica sin servidor.
- [El AWS SAM taller completo](#): un taller diseñado para enseñarle muchas de las principales funciones que AWS SAM ofrece.
- [AWS SAM ejemplos de aplicaciones y patrones](#): ejemplos de aplicaciones y patrones de autores de la comunidad con los que puede seguir experimentando.

Opcional: compruebe la integridad del AWS SAM CLI Instalador de

Al instalar la interfaz de línea de AWS Serverless Application Model comandos (AWS SAMCLI) mediante un instalador de paquetes, puede comprobar su integridad antes de la instalación. Este paso es opcional, pero muy recomendable.

Las dos opciones de verificación que tiene a su disposición son:

- Verifica la firma del archivo instalador.
- Comprueba el valor hash del instalador del paquete.

Cuando esté disponible para tu plataforma, te recomendamos que verifique la opción del archivo de firma. Esta opción ofrece un nivel de seguridad adicional, ya que los valores clave se publican aquí y se administran por separado desde nuestro repositorio GitHub.

Temas

- [Verifica el archivo de la firma del instalador](#)
- [Verifica el valor del hash](#)

Verifica el archivo de la firma del instalador

Linux

arm64: instalador de línea de comandos

AWS SAM usa [GnuPG](#) para firmar AWS SAMCLI el instalador .zip. La verificación se realiza en los siguientes pasos:

1. Utiliza la clave pública principal para verificar la clave pública del firmante.
2. Utiliza la clave pública del firmante para verificar el instalador del paquete AWS SAMCLI.

Para verificar la integridad de la clave pública del firmante

1. Copias la clave pública principal y guárdala en su máquina local como un archivo .txt. Por ejemplo, *primary-public-key.txt*.

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
```

```
Version: GnuPG v2.0.22 (GNU/Linux)
```

```
mQINBGRuSzMBEADsqiw0y78w7F4+sshaMFRIwRGNRM94p5Qey2KMZBxekFtoryVD
D9jE0nvupx4tvhfBHz5EcUHCE0d14MTqdBy6vVAshozgxVb9RE8JpECn5lw7XC69
4Y7Gy1TKKQMEwtDXElkGxIFdUwWjSnPlzfnoXwQYGeE93CUS3h5dImP22Yk1Ct6
eGGhlcbg1X4L8EpFMj7GvcsU8f7ziVI/PyC1Xwy39Q8/I67ip5eU5ddx0/xHqrbL
YC7+8pJPbRMej2twT2LrcpWYAbprMtRoa6WfE0/thoo3xhHpIMHdPfAA86ZNGIN
kRLjGUg7jnPTRW40in3pCc8nT4Tfc1QERkHm641gTC/jUvpmQsM6h/FUVP2i5iE/
JHpJcMuL2Mg6zDo3x+3gTCf+Wqz3rZzxB+wQT3yryZs6efcQy7nR0iRxYBxCSXX0
2cNYzsYLB/bYaW8yqWIHD5IQKhW269gp2E5Khs60zgS3CorMb5/xHgXjUCVgcu8a
a8ncdf9fj13WS5p0ohetPb02ZjWv+MaqrZ0mUIgKbA4RpWZ/fu97P5BW9ylwmIDB
sWy0cMxg8M1vSdLytPieogaM0qMg3u5qXRGBr6Wmevkty0qgnmpGGc5zPiUbt0E8
CnFFqyxBpj5I0nG0KZGVihvn+iRrxrv6G07WW092+Dc6m94U0EEiBR7Qi0wARAQAB
tDRBV1MgU0FNIENMSSBQcm1tYXJ5IDxhd3MtY2FtLWNsaS1wcmltYXJ5QGFtYXpv
bi5jb20+iQI/BBMBCQApBQJkbksZAhsvBQkHhM4ABwsJCAcDAgEGFQgCCQoLBBYC
AwECHgECF4AACgkQQv1fen0tiFqTuhAAzi5+ju5UV0WqHKEv0JS008T4QB8HcqAE
SV03mY6/j29knkcL8ubZP/DbpV7QpHPI2PB5qSXsiDTP3IYPbeY78zHSDjljaIK3
njJLMScFeGPpyfPpwMsuY4nzrRIgAtXShPA8N/k4ZJcafnpNqKj7QnPxiC1KaIQWm
p0tvb8msUF3/s0UTa5Ys/1NRhVC0eGg32ogXGdojZA2kHZWdm9udLo4CDrDcrQT7
NtDcJASapXSQL63XfAS3snEc4e1941YxcjfYZ33rel8K9juyDZfi1slWR/L3AviI
QFIaqSHzy0tP1oinUkoVwL8ThevKD3Ag9CZf1ZLzNCV7yqlF8R1hEZ4zce/3s9E1
WzCFsozb5HfE1AZonmrDh3Sy0EIBM6S6vG5dWnvJrAuSYv2rX38++K5Pr/MIAf0X
D0I1rtA+XDsHNv91SwSy0lt+iClawZAN09IXCiN1r0YcVQlwzDFwCNWDgkwd0qS0
g0A2f8NF91E5nBbeEuYquo011Vy8+ICbg0Fs9LoWZlnVh7/RyY6ssowiU9vGUNHI
```



```
L8f9jqRspIz/Fm3JD86ntZxLVGkeZUz62FqErdohYfkFIVcv7G0NTEyrz5HL1npv
FJ0MR0HjrMrZrn0VZnwBKhpLocTsH+3t5It4ReYEX0f1DIOL/KRwPvjMvBVkXY5
hb1RVDQo0Wc=
=d9oG
-----END PGP PUBLIC KEY BLOCK-----
```

2. Importe la clave pública principal en su conjunto de claves.

```
$ gpg --import primary-public-key.txt

gpg: directory `/home/.../.gnupg' created
gpg: new configuration file `/home/.../.gnupg/gpg.conf' created
gpg: WARNING: options in `/home/.../.gnupg/gpg.conf' are not yet active during this
run
gpg: keyring `/home/.../.gnupg/secring.gpg' created
gpg: keyring `/home/.../.gnupg/pubring.gpg' created
gpg: /home/.../.gnupg/trustdb.gpg: trustdb created
gpg: key 73AD885A: public key "AWS SAM CLI Primary <aws-sam-cli-
primary@amazon.com>" imported
gpg: Total number processed: 1
gpg:          imported: 1 (RSA: 1)
```

3. Copias la clave pública del firmante y guárdala en su máquina local como un archivo .txt. Por ejemplo, *signer-public-key.txt*.

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v2.0.22 (GNU/Linux)

mQINBGgrxIgBEADGCTudveeeVbWpZDGX9Ni57mBRMVSJwQJ6F/PC34jw0DozxTtd
H+ZPsXLvLwerN/DVXBk8E1qNZ5RGptak8j7MPz+MC3n4txibEJpB61vpjJJM+9cC
7whaMLDT/SbykHYXdrnHqa8KsUJ17rPLJcaRN722NSxvYVMI0A9ffVXV7cfEyZi5
MbYF2Gc9LnbKaknImIva7EKe2/wI6YCqC5yytyfWU5dL6oHXsgTnFL9mhziMxv
WhyzawyJG6EJZsJ3WL1bIKApN6XZSxyCx0v1BrebYZjD5v0nA+TJaQ7is8atjtOI
DGe0AViw7k08ChTpjA7YG/Uu7n/Fy7qLF/3Nz0b6cBNjemjBazQ3A3KNCpi5hqFM
Uo1WpoVLr5CXQnc0B3fBUntIoxi0Sk5MKjH9AbYxfqgEX0ZJB9hAlc6LIEy0Yru6
MMBrIHE86IM11NfE/DeLnCdPG23+1Pttyw0t3+9z5QwmPe3VPpEfCySPcdxHKZSP
rLile8qDznEv1PDvQ0qkBXdmVa2yct5VJkdqy6UrN2xa0dpspHjRUjHh/EY/xMt
fwMUj0KohaZ/1pjotCcksAsZWUxCNcFvLYxuxeytVk4F09Es1hj4ihhLUI+43/ic
3DHSEiext7Q8/UccNArkhSCT7U0vvL7QTuP+pjYTyiC8Vx6g/Y5Ht5+qywARAQAB
tDBBV1MgU0FNIENMSSBUZWFtIDxhd3Mtc2FtLWNsaS1zaWduZXJAYW1hem9uLmNv
bT6JAj8EEwEJACKFAMgrxIgCGy8FCQPCZwAHCwkIBwMCAQYVCAIJCgsEFgIDAQIe
AQIXgAAKCRBA1Kuxvt/atJo6EAC/5C8uJs76W5f5V5XNAMzwBFiZuYpop3DRReCo
P68ZZylokAC9ShRZnIOujpDJt1NS7T/G00BzmcpSPkYYE531ALaXcHWmb9XV0Ajg
```

```
J8iboAVBLY0C7mhL/cbJ3v9Q1pXXjyTuhexkJCV8rdHVX/0H8WqTZp1EaRuZ7p8q
PMxddg4ClwstYuhH30/dmNd1Gqfb4Fqy8MnV1yGSXR5Jf+sD1N2U04mbpyk/mr1c
f/jFmx86IkCWJVvdXWCVTe2AFy3NHCdLtdnEvFhokCOQd9wibUWX0j9vq4cVRZT
qamnpAQa0lH3lX0wrjqo8b1AIPoRWSfMtCYvh6kA8MAJv4cAzncXILSLt0E0mzaU
qp5qoy37wNIjeztX6c/q4wss05qTlJhnNu4s3nh5VHultooaYpmDxp+ala5TWeuM
KZDI4KdAGF4z0Raif+N53nd0YIiXkY0goUbsPCnVrCwoK9PjjyoJncq7c14wN150
IQUZEjyYAQDQZqs5XSfY4zW2cCXatrfozKF7R1kSU14DfJwPUyksoNAQEQuezfXyq
kr0gfIWK1r2nMdqS7WgSx/ypS5kdyrHuPZdaYfEVtuezpoT2lQQx0SZqq1p5hI4R
nqmPte53WXJhbC0tgTIJWn+Uy/d5Q/aSIfD6o8gNLS1BDs1j1ku0XKu1sFCHUCZG
aerdsIkCHAQQAQkABgUCaCvFeAAKCRBC/V96c62IWt3/D/9g0LzWtz62lqJRCsri
wcA/yz88ayKb/GUv3FCT5Nd9JZt8y1tW+AE3SPTdcpfZmt5UN2sRzlj061mpKJzp
eBvYQ9og/34ZrRQqeg8bz02u34LKYL1gD0xY0bWtB7TGIxIZZYqZECOPR0Dp6ZzB
abzkRSsJkEk0vbZzJhfWFYs98qfp/G0suFSBE7908Am33DB2jQ/Sol1h1VmNE6Sv
EOgR6+2yEkS2D0+msJMa/V82v9gBTPnxS1NV1d8Dduvt9rbM3LoxiNXUgx/s52yY
U6H3bwUcQ3UY6uRe1UWo5QnMFcDwfg43+q5rmjB4xQyX/BaQyF5K0hZyG+42/pH1
EMw18qN617FTxo3hVQUi/cBah1hQ8EVYsGnHDVxLCisbq5iZvp7+XtmMy1Q417gT
EQRo8feJh31e1GWlccVR2pZgIm1PQ69dzzseHnnKkGhifik0bDGo5/IH2EgI1KFn
SG399RMU/qRz0PLVP3i+zSJmhMqG8cnZaUwE5V4P21vQSc1hhd2Hv/C4SVKNqA2i
+oZbHj2vAkuzTTL075AoANebEjPGqwsKZi5mWUE5Pa931JeiXxWZ1EB7rkgQ1PAB
fsDBhYLt4MxCWAhifLMA6uQ4BhXu2RuX0qNfSbqa8jVF6DB6cD8eAHGpPKfJ0130
LtZnq+n4SfeNbZjD2FQWZR4CrA==
=lHfs
-----END PGP PUBLIC KEY BLOCK-----
```

4. Importe la clave pública del firmante en su conjunto de claves.

```
$ gpg --import signer-public-key.txt
```

```
gpg: key 4094ABB1BEDFDAB4: public key "AWS SAM CLI Team <aws-sam-cli-
signer@amazon.com>" imported
gpg: Total number processed: 1
gpg:             imported: 1 (RSA: 1)
gpg: no ultimately trusted keys found
```

Tome nota del valor clave de la salida. Por ejemplo, **4094ABB1BEDFDAB4**.

5. Utiliza el valor de la clave para obtener y verificar la huella digital de la clave pública del firmante.

```
$ gpg --fingerprint 4094ABB1BEDFDAB4
```

```
pub   rsa4096 2025-05-19 [SCEA] [expires: 2027-05-19]
      EF46 3E86 CA31 933B B688 CC1A 4094 ABB1 BEDF DAB4
uid           [ unknown] AWS SAM CLI Team <aws-sam-cli-signer@amazon.com>
```

La huella digital debe coincidir con la siguiente.

```
EF46 3E86 CA31 933B B688 CC1A 4094 ABB1 BEDF DAB4
```

Si la cadena de huellas digitales no coincide, no usa el instalador de la CLI de AWS SAM. Incorpórate al AWS SAM equipo [creando un problema en el repositorio aws-sam-cli](#). GitHub

6. Verifica las firmas de la clave pública del firmante:

```
$ gpg --check-sigs 4094ABB1BEDFDAB4

pub   rsa4096 2025-05-19 [SCEA] [expires: 2027-05-19]
      EF463E86CA31933BB688CC1A4094ABB1BEDFDAB4
uid           [ unknown] AWS SAM CLI Team <aws-sam-cli-signer@amazon.com>
sig!3        4094ABB1BEDFDAB4 2025-05-19 [self-signature]
sig!         42FD5F7A73AD885A 2025-05-19 AWS SAM CLI Primary <aws-sam-cli-
primary@amazon.com>
```

Si ve 1 signature not checked due to a missing key, repita los pasos anteriores para importar las claves públicas principal y del firmante a su conjunto de claves.

Debería ver en la lista los valores clave de la clave pública principal y de la clave pública del firmante.

Ahora que ha comprobado la integridad de la clave pública del firmante, puede utilizarla para comprobar el instalador del paquete AWS SAMCLI.

Para verificar la integridad del AWS SAM CLI instalador de paquetes

1. Obtener el archivo de firma del paquete de la CLI de AWS SAM: descargue el archivo de firma para el instalador del paquete AWS SAMCLI mediante el siguiente comando:

```
$ wget https://github.com/aws/aws-sam-cli/releases/latest/download/aws-sam-cli-
linux-arm64.zip.sig
```

2. Verifica el archivo de firma: pase los archivos descargados .sig y .zip como parámetros al comando gpg. A continuación, se muestra un ejemplo:

```
$ gpg --verify aws-sam-cli-linux-arm64.zip.sig aws-sam-cli-linux-arm64.zip
```

El resultado debería tener un aspecto similar al siguiente:

```
gpg: Signature made Mon 19 May 2025 01:21:57 AM UTC using RSA key ID
4094ABB1BEDFDAB4
gpg: Good signature from "AWS SAM CLI Team <aws-sam-cli-signer@amazon.com>"
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: EF46 3E86 CA31 933B B688  CC1A 4094 ABB1 BEDF DAB4
```

- Se puede ignorar el mensaje `WARNING: This key is not certified with a trusted signature!`. Esto se produce porque no existe una cadena de confianza entre su clave PGP personal (si es su caso) y la clave PGP de la CLI de AWS SAM . Para obtener más información, consulta [Red de confianza](#).
- Si el resultado incluye la expresión `BAD signature`, verifique si realizó el procedimiento de forma correcta. Si sigues recibiendo esta respuesta, ponte en contacto con el AWS SAM equipo [creando un problema](#) en el GitHub repositorio `aws-sam-cli` y evita usar el archivo descargado.

El mensaje `Good signature from "AWS SAM CLI Team <aws-sam-cli-signer@amazon.com>"` indica que la firma está verificada y que puede continuar con la instalación.

x86_64: instalador de línea de comandos

AWS SAM usa [GnuPG](#) para firmar AWS SAMCLI el instalador .zip. La verificación se realiza en los siguientes pasos:

1. Utiliza la clave pública principal para verificar la clave pública del firmante.
2. Utiliza la clave pública del firmante para verificar el instalador del paquete AWS SAMCLI.

Para verificar la integridad de la clave pública del firmante

1. Copias la clave pública principal y guárdala en su máquina local como un archivo .txt. Por ejemplo, *primary-public-key.txt*.

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v2.0.22 (GNU/Linux)
```

```
mQINBGRuSzMBEADsqiw0y78w7F4+sshaMFRIwRGNRm94p5Qey2KMZBxekFtoryVD
D9jE0nvupx4tvhfBH5EcUHCe0d14MTqdBy6vVAshozgxVb9RE8JpECn5lw7XC69
4Y7Gy1TKKQMEWtDXElkGxIFdUWvWjSnPlzfnoXwQYGeE93CUS3h5dImP22Yk1Ct6
eGGhlcbg1X4L8EpFMj7GvcsU8f7ziVI/PyC1Xwy39Q8/I67ip5eU5ddx0/xHqrbL
YC7+8pJPbRMej2twT2LrcpWYAbprMtRoa6WfE0/thoo3xhHpIMHdPfAA86ZNGIN
kRLjGUg7jnPTRW40in3pCc8nT4Tfc1QERkHm641gTC/jUvpmQsM6h/FUVP2i5iE/
JHpJcMuL2Mg6zDo3x+3gTCf+Wqz3rZzxB+wQT3yryZs6efcQy7nR0iRxYBxCSXX0
2cNYzsYLB/bYaW8yqWIHD5IqKhw269gp2E5Khs60zgS3CorMb5/xHgXjUCVgcu8a
a8ncdf9fj13WS5p0ohetPb02ZjWv+MaqrZ0mUIgKbA4RpWZ/fU97P5BW9ylwmIDB
sWy0cMxg8M1vSdLytPieogaM0qMg3u5qXRGBr6WmevktY0qgnmpGGc5zPiUbtOE8
CnFFqyxBpj5IOng0KZGVihvn+iRrxr6G07Ww092+Dc6m94U0EEiBR7Qi0wARAQAB
tDRBV1MgU0FNIENMSSBQcm1tYXJ5IDxhd3Mt2FtLWNsaS1wcm1tYXJ5QGFtYXpv
bi5jb20+iQI/BBMBCQApBQJkbksZAhsvBQkHhM4ABwsJCACDAgEGFQgCCQoLBBYC
AwECHgECF4AACgkQQv1fen0tiFqTuhAAzi5+ju5UV0WqHKEvOJS008T4QB8HcqAE
SV03mY6/j29knkcL8ubZP/DbpV7QpHPi2PB5qSXsiDTP3IYPbeY78zHSDjljaIK3
njJLMScFeGPYfPpwMsuY4nzrRIgAtXShPA8N/k4ZJcafnpNqKj7QnPxiC1KaIQWm
p0tvb8msUF3/s0UTa5Ys/1NRhVC0eGg32ogXGdojZA2kHZWdm9udLo4CDrDcrQT7
NtDcJASapXSQL63XfAS3snEc4e1941YxcjFYZ33rel8K9juyDZfi1slWR/L3AviI
QFIaqSHzy0tPloinUkoVwL8ThevKD3Ag9CZf1ZLzNCV7yqlF8RlhEZ4zcE/3s9E1
WzCFsozb5HfE1AZonmrDh3Sy0EIBMCS6vG5dWnvJrAuSYv2rX38++K5Pr/MIAf0X
D0I1rtA+XDshNv9lSwSy0lt+iClawZAN09IXCiN1r0YcVQlwzDFwCNWDgkwd0qS0
g0A2f8NF91E5nBbeEuYquo011Vy8+ICbg0Fs9LoWZlnVh7/RyY6ssowiU9vGUNHI
L8f9jqRspIz/Fm3JD86ntZxLVGkeZUz62FqErdohYfkFIVcv7G0NTEyrz5HL1npv
FJ0MR0HjrMrZrn0VZnwBKhpBLocTsH+3t5It4ReYEX0f1DIOL/KRwPvjMvBVkXY5
hb1RVDQo0Wc=
=d9oG
-----END PGP PUBLIC KEY BLOCK-----
```

2. Importe la clave pública principal en su conjunto de claves.

```
$ gpg --import primary-public-key.txt

gpg: directory `/home/.../.gnupg' created
gpg: new configuration file `/home/.../.gnupg/gpg.conf' created
gpg: WARNING: options in `/home/.../.gnupg/gpg.conf' are not yet active during this
run
gpg: keyring `/home/.../.gnupg/secring.gpg' created
gpg: keyring `/home/.../.gnupg/pubring.gpg' created
gpg: /home/.../.gnupg/trustdb.gpg: trustdb created
gpg: key 73AD885A: public key "AWS SAM CLI Primary <aws-sam-cli-
primary@amazon.com>" imported
gpg: Total number processed: 1
```

```
gpg: imported: 1 (RSA: 1)
```

3. Copias la clave pública del firmante y guárdala en su máquina local como un archivo `.txt`. Por ejemplo, *signer-public-key.txt*.

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v2.0.22 (GNU/Linux)

mQINBGgrxIgBEADGCTudveeeVbWpZDGX9Ni57mBRMVSJwQJ6F/PC34jw0DozxTtd
H+ZPsXLvLwerN/DVXbK8E1qNZ5RGptak8j7MPz+MC3n4txibEJpB61vpjJJM+9cC
7whaMLDT/SbykHYXdrnHqa8KsUJl7rPLJcaRN722NSxvYVMI0A9ffVXV7cfEyZi5
MbYF2Gc9LNBKaknImIva7EKeeh2/wI6YCqC5yytyfWU5dL6oHXsgTnFL9mhziMxv
WhyzawyJG6EJZsJ3Wl1bIKApN6XZSXYCx0v1BrebYZjD5v0nA+TJaQ7is8atjt0I
DGe0AViw7k08ChTpjA7YG/Uu7n/Fy7qLF/3Nz0b6cBNjemjBazQ3A3KNCpi5hqFM
Uo1WpoVlr5CXQnc0B3fBUntIoxi0Sk5MKjH9AbYxfqgEX0ZJB9hAlc6LIEy0Yru6
MMBrIHE86IM11nFE/DeLnCdPG23+1Pttwy0t3+9z5QwmPe3VPpEfCySPcdxHKZSP
rLile8qDznEv1PDvQ0qkBXdmTva2yct5VJkdqy6UrN2xa0dpspHjRUjHh/EY/xMt
fwMUj0KohaZ/1pjotCcksAsZWUxCNcFvLYxuxeytVk4F09Es1hj4ihhLUI+43/ic
3DHSEiext7Q8/UccNArkhSCT7U0vvL7QTuP+pjYTyiC8Vx6g/Y5Ht5+qywARAQAB
tDBBV1MgU0FNIENMSSBUZWFtIDxhd3Mtc2FtLWNsaS1zaWduZXJAYW1hem9uLmNv
bT6JAj8EEwEJACKFAMgrxIgCGy8FCQPCZwAHCwkIBwMCAQYVCAIJCgsEFgIDAQIE
AQIXgAAKCRBA1Kuxvt/atJo6EAC/5C8uJs76W5f5V5XNAMzwBFiZuYpop3DRReCo
P68ZZyl0kAC9ShRZnIOujpDJt1NS7T/G00BzmcpspkYYE531ALaXcHWmb9XV0Ajg
J8iboAVBLY0C7mhL/cbJ3v9Q1pXXjyTuhexkJCV8rdHVX/0H8WqTZp1EaRuZ7p8q
PMxddg4ClwstYuh30/dmNd1Gqfb4Fqy8MnV1yGSXR55Jf+sD1N2U04mbpyk/mr1c
f/jFmx86IkCWJVvdXWCVTe2AFy3NHCdLtdnEvFhokC0Qd9wibUWX0j9vq4cVRZT
qamnpAQa0lH3lX0wrjqo8b1AIPoRWSfMtCYvh6kA8MAJv4cAznzXILSLt0E0mzaU
qp5qoy37wNIjeztX6c/q4wss05qTlJhnNu4s3nh5VHu1tooaYpmDxp+ala5TweuM
KZDI4KdAGF4z0Raif+N53nd0YIiXkY0goUbsPCnVrCwoK9PjjyoJncq7c14wN150
IQUZEjyYAQDGZqs5XSfY4zW2cCXatrfzKF7R1kSU14DfJwPUyksoNAQEQuezfXyq
kr0gfIWK1r2nMdqS7WgSx/ypS5kdyrHuPZdaYfEVtuezpoT2lQQx0SZqq1p5hI4R
nqmPte53WXJhbC0tgTIJWn+Uy/d5Q/aSiFD6o8gNLS1BDs1j1ku0XKu1sFCHUcZG
aerdsIkCHAQQAQkABgUCaCvFeAAKCRBC/V96c62IWt3/D/9g0LzWtz62lqJRCsri
wcA/yz88ayKb/GUv3FCT5Nd9JZt8y1tW+AE3SPTdcpfZmt5UN2sRzlJ061mpKJzp
eBvYQ9og/34ZrRQqeg8bz02u34LKYl1gD0xY0bWtB7TGIXIZZYqZECOPR0Dp6ZzB
abzkRSsJkEk0vbZzJhfWfYs98qfp/G0suFSBE7908Am33DB2jQ/Sol1h1VmNE6Sv
EOgR6+2yEkS2D0+msJMa/V82v9gBTPnxS1NV1d8Dduvt9rbM3LoxiNXUgx/s52yY
U6H3bwUcQ3UY6uRe1UWo5QnMFCdWfg43+q5rmjB4xQyX/BaQyF5K0hZyG+42/pH1
EMw18qN617FTxo3hvQui/cBah1hQ8EVYsGnHDVxLCisbq5iZvp7+XtmMy1Q417gT
EQRo8feJh31e1GWlccVR2pZgIm1PQ69dzzseHnnKkGhifik0bDGo5/IH2EgI1KFn
SG399RMU/qRz0PLVP3i+zSJmhMqG8cnZaUwE5V4P21vQSc1hhd2Hv/C4SVKNqA2i
+oZbHj2vAkuzTTL075AoANebEjPGqwsKZi5mWUE5Pa931JeixxWZ1EB7rkgQ1PAB
fsDBhYLt4MxCWAhifLMA6uQ4BhXu2RuX0qNfSbqa8jVF6DB6cD8eAHGpPKfJ0130
```

```

LtZnq+n4SfeNbZjD2FQWZR4CrA==
=1Hfs
-----END PGP PUBLIC KEY BLOCK-----

```

4. Importe la clave pública del firmante en su conjunto de claves.

```

$ gpg --import signer-public-key.txt

gpg: key 4094ABB1BEDFDAB4: public key "AWS SAM CLI Team <aws-sam-cli-
signer@amazon.com>" imported
gpg: Total number processed: 1
gpg:             imported: 1 (RSA: 1)
gpg: no ultimately trusted keys found

```

Tome nota del valor clave de la salida. Por ejemplo, **4094ABB1BEDFDAB4**.

5. Utiliza el valor de la clave para obtener y verificar la huella digital de la clave pública del firmante.

```

$ gpg --fingerprint 4094ABB1BEDFDAB4

pub   rsa4096 2025-05-19 [SCEA] [expires: 2027-05-19]
      EF46 3E86 CA31 933B B688  CC1A 4094 ABB1 BEDF DAB4
uid           [ unknown] AWS SAM CLI Team <aws-sam-cli-signer@amazon.com>

```

La huella digital debe coincidir con la siguiente.

```
EF46 3E86 CA31 933B B688  CC1A 4094 ABB1 BEDF DAB4
```

Si la cadena de huellas digitales no coincide, no usa el instalador de la CLI de AWS SAM. Incorpórate al AWS SAM equipo [creando un problema en el repositorio aws-sam-cli](#). GitHub

6. Verifica las firmas de la clave pública del firmante:

```

$ gpg --check-sigs 4094ABB1BEDFDAB4

pub   rsa4096 2025-05-19 [SCEA] [expires: 2027-05-19]
      EF463E86CA31933BB688CC1A4094ABB1BEDFDAB4
uid           [ unknown] AWS SAM CLI Team <aws-sam-cli-signer@amazon.com>
sig!3        4094ABB1BEDFDAB4 2025-05-19  [self-signature]
sig!         42FD5F7A73AD885A 2025-05-19 AWS SAM CLI Primary <aws-sam-cli-
primary@amazon.com>

```

Si ve `1 signature not checked due to a missing key`, repita los pasos anteriores para importar las claves públicas principal y del firmante a su conjunto de claves.

Debería ver en la lista los valores clave de la clave pública principal y de la clave pública del firmante.

Ahora que ha comprobado la integridad de la clave pública del firmante, puede utilizarla para comprobar el instalador del paquete AWS SAMCLI.

Para verificar la integridad del AWS SAM CLI instalador de paquetes

1. Obtener el archivo de firma del paquete de la CLI de AWS SAM: descargue el archivo de firma para el instalador del paquete AWS SAMCLI mediante el siguiente comando:

```
$ wget https://github.com/aws/aws-sam-cli/releases/latest/download/aws-sam-cli-linux-x86_64.zip.sig
```

2. Verifica el archivo de firma: pase los archivos descargados `.sig` y `.zip` como parámetros al comando `gpg`. A continuación, se muestra un ejemplo:

```
$ gpg --verify aws-sam-cli-linux-x86_64.zip.sig aws-sam-cli-linux-x86_64.zip
```

El resultado debería tener un aspecto similar al siguiente:

```
gpg: Signature made Mon 19 May 2025 01:21:57 AM UTC using RSA key ID
4094ABB1BEDFDAB4
gpg: Good signature from "AWS SAM CLI Team <aws-sam-cli-signer@amazon.com>"
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: EF46 3E86 CA31 933B B688 CC1A 4094 ABB1 BEDF DAB4
```

- Se puede ignorar el mensaje `WARNING: This key is not certified with a trusted signature!`. Esto se produce porque no existe una cadena de confianza entre su clave PGP personal (si es su caso) y la clave PGP de la CLI de AWS SAM. Para obtener más información, consulta [Red de confianza](#).
- Si el resultado incluye la expresión `BAD signature`, verifique si realizó el procedimiento de forma correcta. Si sigues recibiendo esta respuesta, ponte en contacto con el AWS SAM

equipo [creando un problema](#) en el GitHub repositorio aws-sam-cli y evita usar el archivo descargado.

El mensaje `Good signature from "AWS SAM CLI Team <aws-sam-cli-signer@amazon.com>"` indica que la firma está verificada y que puede continuar con la instalación.

macOS

Instalador de interfaz gráfica de usuario y línea de comandos

Puede comprobar la integridad del archivo de firmas del instalador del paquete de la CLI de AWS SAM mediante la herramienta `pkgutil` o manualmente.

Para verificarlo, utilice `pkgutil`

1. Ejecuta el siguiente comando, que proporciona la ruta al instalador descargado en la máquina local:

```
$ pkgutil --check-signature /path/to/aws-sam-cli-installer.pkg
```

A continuación, se muestra un ejemplo:

```
$ pkgutil --check-signature /Users/user/Downloads/aws-sam-cli-macos-arm64.pkg
```

2. En la salida, localice el SHA256 fingerprint para Developer ID Installer: AMZN Mobile LLC. A continuación, se muestra un ejemplo:

```
Package "aws-sam-cli-macos-arm64.pkg":
  Status: signed by a developer certificate issued by Apple for distribution
  Notarization: trusted by the Apple notary service
  Signed with a trusted timestamp on: 2026-01-28 07:39:16 +0000
  Certificate Chain:
    1. Developer ID Installer: AMZN Mobile LLC (94KV3E626L)
      Expires: 2030-09-26 00:18:06 +0000
      SHA256 Fingerprint:
        5C 45 BE 63 FD 52 10 07 2D 66 56 77 5C A9 FF 25 91 6D 3F 01 F7 0E
        9A 8A 05 F6 2D 62 B2 88 8D A9
      -----
    2. Developer ID Certification Authority
```

```
Expires: 2031-09-17 00:00:00 +0000
SHA256 Fingerprint:
    F1 6C D3 C5 4C 7F 83 CE A4 BF 1A 3E 6A 08 19 C8 AA A8 E4 A1 52 8F
    D1 44 71 5F 35 06 43 D2 DF 3A
-----
```

3. Apple Root CA

```
Expires: 2035-02-09 21:40:36 +0000
SHA256 Fingerprint:
    B0 B1 73 0E CB C7 FF 45 05 14 2C 49 F1 29 5E 6E DA 6B CA ED 7E 2C
    68 C5 BE 91 B5 A1 10 01 F0 24
```

3. Developer ID Installer: AMZN Mobile LLC SHA256 fingerprint debe coincidir con el siguiente valor:

```
5C 45 BE 63 FD 52 10 07 2D 66 56 77 5C A9 FF 25 91 6D 3F 01 F7 0E 9A 8A 05 F6 2D 62
B2 88 8D A9
```

Si la cadena de huellas digitales no coincide, no usa el instalador de la CLI de AWS SAM. [Crea](#) un problema en el repositorio AWS SAM `aws-sam-cli` para comunicarte con el equipo. GitHub Si la cadena de huellas digitales coincide, puede continuar con el instalador del paquete.

Verifica el instalador del paquete manualmente

- Consulte [Cómo verificar la autenticidad de las actualizaciones de software de Apple descargadas manualmente](#) en el sitio web de soporte de Apple.

Windows

El AWS SAMCLI instalador está empaquetado como MSI archivos para el sistema operativo.

Windows

Comprueba la integridad del instalador del paquete

1. Right-click en el instalador y abra la ventana Propiedades.
2. Elija la pestaña Firmas digitales.
3. En la Lista de firmas, elija Amazon Services, Inc. y, a continuación, Detalles.
4. Elija la pestaña General si aún no lo ha hecho, y luego elija Ver certificado.
5. Elija la pestaña Detalles y luego elija Todos en la lista desplegable Mostrar, si aún no está seleccionada.

- Desplácese hacia abajo hasta que vea el campo Huella digital y, a continuación, seleccione Huella digital. Así se muestra el valor completo de la huella digital en la ventana inferior.
- Haga coincidir el valor de la huella digital con el siguiente valor. Si el valor coincide, continúe con la instalación. Si no es así, diríjase al AWS SAM equipo [creando un problema](#) en el repositorio GitHub aws-sam-cli.

```
6c 4e f0 59 67 db 0a 2c 69 87 17 31 ec 9e 51 5d 7c d6 e9 52
```

Verifica el valor del hash

Linux

x86_64: instalador de línea de comandos

Comprueba la integridad y autenticidad de los archivos de instalación descargados mediante la generación de un valor hash mediante el siguiente comando:

```
$ sha256sum aws-sam-cli-linux-x86_64.zip
```

El resultado debe tener el siguiente aspecto:

```
<64-character SHA256 hash value> aws-sam-cli-linux-x86_64.zip
```

[Compare el valor SHA-256 hash de 64 caracteres con el de la versión que desee AWS SAMCLI en las notas de la versión que aparecen en la página.AWS SAMCLI](#) GitHub

macOS

Instalador de interfaz gráfica de usuario y línea de comandos

Comprueba la integridad y autenticidad del instalador descargado mediante la generación de un valor hash mediante el siguiente comando:

```
$ shasum -a 256 path-to-pkg-installer/name-of-pkg-installer

# Examples
$ shasum -a 256 ~/Downloads/aws-sam-cli-macos-arm64.pkg
$ shasum -a 256 ~/Downloads/aws-sam-cli-macos-x86_64.pkg
```

[Compara tu valor SHA-256 hash de 64 caracteres con el valor correspondiente del repositorio de notas de la AWS SAMCLI versión.](#) [GitHub](#)

Tutorial: Implemente una aplicación Hello World con AWS SAM

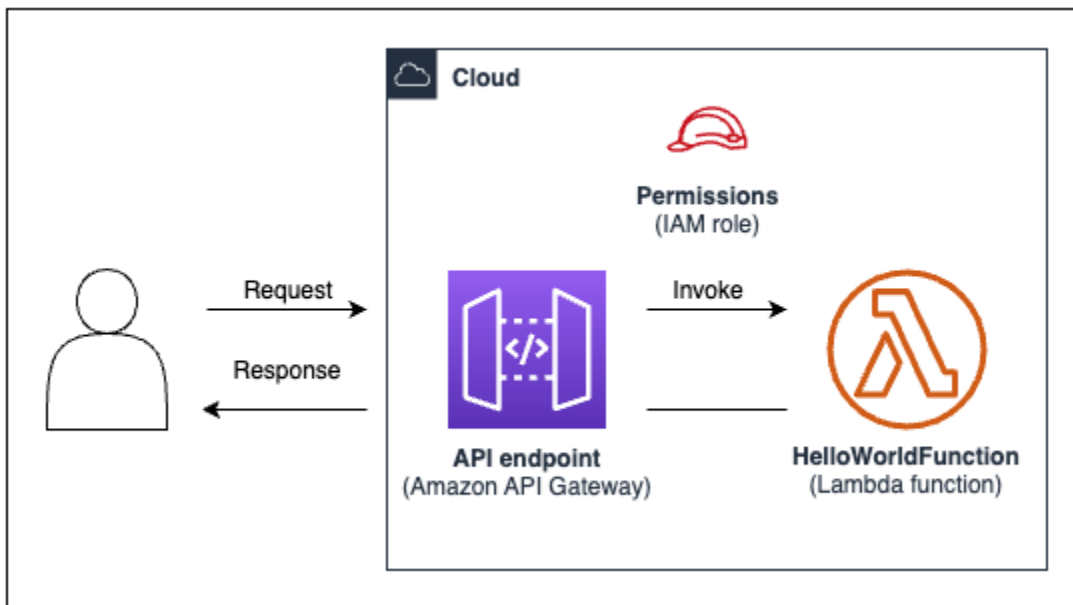
En este tutorial, utilizará la interfaz de línea de AWS Serverless Application Model comandos (AWS SAMCLI) para realizar lo siguiente:

- Inicializa, crea e implementa una aplicación de muestra Hello World
- Realice cambios locales y sincronícelos con AWS CloudFormation.
- Haga pruebas locales en el host de desarrollo.
- Elimina la aplicación de muestra de la Nube de AWS.

La aplicación de muestra Hello World implementa un backend de API básico. Consta de los recursos siguientes:

- Amazon API Gateway: punto de conexión de la API que utilizarás para invocar tu función.
- AWS Lambda: función que procesa la solicitud HTTP API GET y devuelve el mensaje `hello world`.
- AWS Identity and Access Management Función (IAM): proporciona permisos para que los servicios interactúen de forma segura.

El siguiente diagrama muestra los componentes de esta aplicación:



Temas

- [Requisitos previos](#)
- [Paso 1: inicializar la aplicación de muestra Hello World](#)
- [Paso 2: crear tu propia aplicación](#)
- [Paso 3: Implemente la aplicación en Nube de AWS](#)
- [Paso 4: ejecutar la aplicación](#)
- [Paso 5: Interactúa con tu función en el Nube de AWS](#)
- [Paso 6: Modifique y sincronice su aplicación con Nube de AWS](#)
- [Paso 7: \(opcional\) probar tu aplicación localmente](#)
- [Paso 8: Elimine la aplicación del Nube de AWS](#)
- [Resolución de problemas](#)
- [Más información](#)

Requisitos previos

Asegúrate de haber completado lo siguiente:

- [AWS SAM requisitos previos](#)
- [Instala la AWS SAM CLI](#)

 Note

Si prefiere usar Visual Studio Code para el desarrollo sin servidor, consulte [Creación de aplicaciones sin servidor](#) en la Guía del usuario del AWS Toolkit for VS Code para obtener instrucciones paso a paso sobre el uso de la interfaz VS Code.

Paso 1: inicializar la aplicación de muestra Hello World

En este paso, utilizarás la AWS SAM CLI para crear un proyecto de aplicación de muestra Hello World en su unidad local.

Para inicializar la aplicación de muestra Hello World

1. En la línea de comandos, ejecuta lo siguiente desde el directorio de inicio que desees:

```
$ sam init
```

 Note

Este comando inicializa la aplicación sin servidor y crea el directorio del proyecto. El directorio contiene varios archivos y carpetas, incluido el `template.yaml` archivo, que es su AWS SAM plantilla.

La versión de Python instalada debe coincidir con la propiedad `Runtime` especificada en `template.yaml`. Para las cargas de trabajo de producción, recomendamos usar Python 3.12 o versiones anteriores que sean totalmente compatibles con los entornos AWS. Si las versiones de Python no coinciden, se producirán errores de compilación. Para resolver problemas de compatibilidad de versiones:

- Especifique un [tiempo de ejecución](#) compatible durante la inicialización: `sam init --runtime python3.9`
- Modifique la propiedad `Runtime` en `template.yaml` después de la inicialización

2. La AWS SAM CLI te guiará a través de la inicialización de una nueva aplicación. Configura lo siguiente:
 1. Selecciona Plantillas de inicio rápido de AWS para elegir una plantilla de inicio.
 2. Selecciona la plantilla Ejemplo de Hello World y descárgala.

3. Usa el tiempo de ejecución de Python y el tipo de paquete zip.
4. Para este tutorial, opte por no participar en el rastreo. AWS X-Ray Para obtener más información, consulte [¿Qué es? AWS X-Ray](#) en la Guía para AWS X-Ray desarrolladores.
5. Para este tutorial, opte por no monitorizar con Amazon CloudWatch Application Insights. Para obtener más información, consulte [Amazon CloudWatch Application Insights](#) en la Guía del CloudWatch usuario de Amazon.
6. Para este tutorial, debe desactivar el parámetro de registro estructurado en formato JSON en las funciones de Lambda.
7. Asigna el nombre sam-app a tu aplicación.

Para usar el flujo interactivo de la CLI de AWS SAM:

- Los corchetes ([]) indican los valores predeterminados. Deja la respuesta en blanco para seleccionar el valor predeterminado.
- Escribe **y** para sí y **n** para no.

A continuación, se muestra un ejemplo de flujo interactivo con `sam init`.

```
$ sam init
...
Which template source would you like to use?
  1 - AWS Quick Start Templates
  2 - Custom Template Location
Choice: 1

Choose an AWS Quick Start application template
  1 - Hello World Example
  2 - Data processing
  3 - Hello World Example with Powertools for AWS Lambda
  4 - Multi-step workflow
  5 - Scheduled task
  6 - Standalone function
  7 - Serverless API
  8 - Infrastructure event management
  9 - Lambda Response Streaming
 10 - GraphQLApi Hello World Example
 11 - Full Stack
 12 - Lambda EFS example
```

- 13 - Serverless Connector Hello World Example
- 14 - Multi-step workflow with Connectors
- 15 - DynamoDB Example
- 16 - Machine Learning

Template: **1**

Use the most popular runtime and package type? (Python and zip) [y/N]: **y**

Would you like to enable X-Ray tracing on the function(s) in your application? [y/N]: **ENTER**

Would you like to enable monitoring using CloudWatch Application Insights?
For more info, please view <https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/cloudwatch-application-insights.html> [y/N]: **ENTER**

Would you like to set Structured Logging in JSON format on your Lambda functions?
[y/N]: **ENTER**

Project name [sam-app]: **ENTER**

3. **AWS SAMCLID** Descarga la plantilla inicial y crea la estructura de directorios del proyecto de la aplicación en su máquina local. A continuación, se muestra un ejemplo de resultados del AWS SAM CLI output.

Cloning from <https://github.com/aws/aws-sam-cli-app-templates> (process may take a moment)

Generating application:

Name: sam-app
Runtime: python3.9
Architectures: x86_64
Dependency Manager: pip
Application Template: hello-world
Output Directory: .
Configuration file: sam-app/samconfig.toml

Next steps can be found in the README file at sam-app/README.md

Commands you can use next
=====


```
[*] Create pipeline: cd sam-app && sam pipeline init --bootstrap
[*] Validate SAM template: cd sam-app && sam validate
[*] Test Function in the Cloud: cd sam-app && sam sync --stack-name {stack-name} --
watch
```

4. Desde la línea de comandos, dirígete al directorio `sam-app` recién creado. A continuación se muestra un ejemplo de lo que la CLI de AWS SAM ha creado:

 Note

Si el comando `tree` no se instala automáticamente, ejecute este comando: `brew install tree`

```
$ cd sam-app

$ tree

### README.md
### __init__.py
### events
#   ### event.json
### hello_world
#   ### __init__.py
#   ### app.py
#   ### requirements.txt
### samconfig.toml
### template.yaml
### tests
    ### __init__.py
    ### integration
    #   ### __init__.py
    #   ### test_api_gateway.py
    ### requirements.txt
    ### unit
        ### __init__.py
        ### test_handler.py

6 directories, 14 files
```

Algunos archivos importantes que hay que destacar:

- `hello_world/app.py`: contiene el código de tu función de Lambda.
- `hello_world/requirements.txt`: contiene cualquier dependencia de Python que tu función de Lambda necesite.
- `samconfig.toml`— Archivo de configuración para su aplicación que almacena los parámetros predeterminados utilizados por AWS SAMCLI.
- `template.yaml`— La AWS SAM plantilla que contiene el código de infraestructura de la aplicación.

¡Ya tienes una aplicación sin servidor creada enteramente en tu equipo local!

Paso 2: crear tu propia aplicación

En este paso, usa AWS SAM CLI para crear la aplicación y prepararla para la implementación. Mientras la está creando, la de AWS SAM CLI crea un directorio de `.aws-sam` y organiza allí las dependencias de las funciones, el código del proyecto y los archivos del proyecto.

Para crear tu aplicación

- En tu línea de comandos, desde el directorio del proyecto `sam-app`, ejecuta lo siguiente:

```
$ sam build
```

Note

Si no tienes Python en tu equipo local, usa el comando `sam build --use-container` en su lugar. La AWS SAM CLI creará un contenedor de Docker que incluye el tiempo de ejecución y las dependencias de tu función. Este comando requiere que tengas Docker en tu equipo local. Para instalar Docker, consulta [Instalar Docker](#).

A continuación, se muestra un ejemplo de resultados de la CLI de AWS SAM.

```
$ sam build
Starting Build use cache
Manifest file is changed (new hash: 3298f1304...d4d421) or dependency folder (.aws-sam/deps/4d3dfad6-a267-47a6-a6cd-e07d6fae318c) is missing for (HelloWorldFunction), downloading dependencies and copying/building source
```

```
Building codeuri: /Users/.../Demo/sam-tutorial1/sam-app/hello_world runtime:
python3.9 metadata: {} architecture: x86_64 functions: HelloWorldFunction
Running PythonPipBuilder:CleanUp
Running PythonPipBuilder:ResolveDependencies
Running PythonPipBuilder:CopySource
Running PythonPipBuilder:CopySource
```

Build Succeeded

```
Built Artifacts   : .aws-sam/build
Built Template    : .aws-sam/build/template.yaml
```

Commands you can use next

=====

```
[*] Validate SAM template: sam validate
[*] Invoke Function: sam local invoke
[*] Test Function in the Cloud: sam sync --stack-name {{stack-name}} --watch
[*] Deploy: sam deploy --guided
```

El siguiente es un ejemplo abreviado del directorio de `.aws-sam` creado por la CLI de AWS SAM :

```
.aws-sam
### build
#   ### HelloWorldFunction
#   #   ### __init__.py
#   #   ### app.py
#   #   ### requirements.txt
#   ### template.yaml
### build.toml
```

Algunos archivos importantes que hay que destacar:

- `build/HelloWorldFunction`: contiene el código y las dependencias de tu función de Lambda. La AWS SAM CLI crea un directorio para cada función de tu aplicación.
- `build/template.yaml`— Contiene una copia de la AWS SAM plantilla a la que se hace referencia CloudFormation en el momento de la implementación.
- `build.toml`: archivo de configuración que almacena los valores de los parámetros predeterminados a los que hace referencia la AWS SAM CLI al crear e implementar la aplicación.

Ahora estás listo o lista para implementar la aplicación en la Nube de AWS.

Paso 3: Implemente la aplicación en Nube de AWS

Note

Este paso requiere la configuración de AWS las credenciales. Para obtener más información, consulta [Paso 2: utilice el AWS CLI para configurar AWS credentials](#) en [AWS SAM requisitos previos](#).

En este paso, utiliza la AWS SAM CLI para implementar la aplicación en la Nube de AWS. AWS SAMCLIHará lo siguiente:

- Guiarte a través de la configuración de los ajustes de la aplicación para la implementación.
- Cargar tus archivos de aplicación en Amazon Simple Storage Service (Amazon S3).
- Transforma tu AWS SAM plantilla en una CloudFormation plantilla. A continuación, carga la plantilla en el CloudFormation servicio para aprovisionar AWS los recursos.

Para implementar la aplicación

1. En tu línea de comandos, desde el directorio del proyecto sam-app, ejecuta lo siguiente:

```
$ sam deploy --guided
```

2. Sigue el flujo interactivo de la CLI de AWS SAM para configurar los ajustes de la aplicación. Configura lo siguiente:
 1. El nombre de la CloudFormation pila: una pila es un conjunto de AWS recursos que puedes gestionar como una sola unidad. Para obtener más información, consulta [Trabajar con pilas](#) en la Guía del usuario de AWS CloudFormation .
 2. El Región de AWS lugar en el que desplegar tu CloudFormation pila. Para obtener más información, consulta [CloudFormation Puntos de conexión de](#) en la AWS CloudFormation Guía de usuario.
 3. Para este tutorial, desactiva confirmar los cambios antes de la implementación.

4. Permitir la creación de funciones de IAM: permite AWS SAM crear la función de IAM necesaria para que el recurso de API Gateway y el recurso de función de Lambda interactúen.
5. Para este tutorial, desactiva deshabilitar la reversión.
6. Permitir HelloWorldFunction sin autorización definida: este mensaje se muestra porque su punto final de API Gateway está configurado para ser de acceso público, sin autorización. Dado que esta es la configuración prevista para tu aplicación Hello World, deje que la CLI de AWS SAM continúe. Para obtener más información acerca de la autorización de configuración, consulta [Controle el acceso a la API con su AWS SAM plantilla](#).
7. Guardar los argumentos en el archivo de configuración: esto actualizará el archivo `samconfig.toml` de la aplicación con tus preferencias de implementación.
8. Selecciona el nombre del archivo de configuración predeterminado.
9. Selecciona el entorno de configuración predeterminado.

A continuación se muestra un ejemplo del flujo interactivo `sam deploy --guided`:

```
$ sam deploy --guided
```

```
Configuring SAM deploy
```

```
=====
```

```
Looking for config file [samconfig.toml] : Found
```

```
Reading default arguments : Success
```

```
Setting default arguments for 'sam deploy'
```

```
=====
```

```
Stack Name [sam-app]: ENTER
```

```
AWS Region [us-west-2]: ENTER
```

```
#Shows you resources changes to be deployed and require a 'Y' to initiate  
deploy
```

```
Confirm changes before deploy [Y/n]: n
```

```
#SAM needs permission to be able to create roles to connect to the resources in  
your template
```

```
Allow SAM CLI IAM role creation [Y/n]: ENTER
```

```
#Preserves the state of previously provisioned resources when an operation  
fails
```

```
Disable rollback [y/N]: ENTER
```

```
HelloWorldFunction may not have authorization defined, Is this okay? [y/N]: y
```

```
Save arguments to configuration file [Y/n]: ENTER
```

```
SAM configuration file [samconfig.toml]: ENTER
SAM configuration environment [default]: ENTER
```

3. AWS SAMCLIDespliega la aplicación de la siguiente manera:

- La AWS SAM CLI crea un bucket de Amazon S3 y carga el directorio `.aws-sam`.
- AWS SAMCLITransforma la AWS SAM plantilla en ella CloudFormation y la carga en el CloudFormation servicio.
- CloudFormation aprovisiona sus recursos.

Durante la implementación, la AWS SAM CLI muestra tu progreso. El siguiente es un ejemplo de output:

```
Looking for resources needed for deployment:
```

```
Managed S3 bucket: aws-sam-cli-managed-default-samclisam-s3-demo-
bucket-1a4x26zbcdkqr
```

```
A different default S3 bucket can be set in samconfig.toml
```

```
Parameter "stack_name=sam-app" in [default.deploy.parameters] is defined as a
global parameter [default.global.parameters].
```

```
This parameter will be only saved under [default.global.parameters] in /
Users/.../Demo/sam-tutorial1/sam-app/samconfig.toml.
```

```
Saved arguments to config file
```

```
Running 'sam deploy' for future deployments will use the parameters saved
above.
```

```
The above parameters can be changed by modifying samconfig.toml
```

```
Learn more about samconfig.toml syntax at
```

```
https://docs.aws.amazon.com/serverless-application-model/latest/developerguide/
serverless-sam-cli-config.html
```

```
File with same data already exists at sam-app/da3c598813f1c2151579b73ad788cac8,
skipping upload
```

```
Deploying with following values
```

```
=====
```

```
Stack name           : sam-app
Region               : us-west-2
Confirm changeset    : False
Disable rollback     : False
```

```

    Deployment s3 bucket      : aws-sam-cli-managed-default-samclisam-s3-demo-
bucket-1a4x26zbcdkqr
    Capabilities              : ["CAPABILITY_IAM"]
    Parameter overrides       : {}
    Signing Profiles          : {}

```

Initiating deployment

=====

File with same data already exists at sam-
app/2bebf67c79f6a743cc5312f6dfc1efee.template, skipping upload

Waiting for changeset to be created..

CloudFormation stack changeset

Operation	LogicalResourceId
ResourceType	Replacement
* Modify	HelloWorldFunction
AWS::Lambda::Function	False
* Modify	ServerlessRestApi
AWS::ApiGateway::RestApi	False
- Delete	AwsSamAutoDependencyLayerNestedSt
AWS::CloudFormation::Stack	N/A
	ack

Changeset created successfully. arn:aws:cloudformation:us-
west-2:012345678910:changeSet/samcli-deploy1678917603/22e05525-08f9-4c52-a2c4-
f7f1fd055072

2023-03-15 12:00:16 - Waiting for stack create/update to complete

CloudFormation events from stack operations (refresh every 0.5 seconds)

ResourceStatus	ResourceType
LogicalResourceId	ResourceStatusReason
UPDATE_IN_PROGRESS	AWS::Lambda::Function
HelloWorldFunction	-

```

UPDATE_COMPLETE                               AWS::Lambda::Function
  HelloWorldFunction                           -
UPDATE_COMPLETE_CLEANUP_IN_PROGRE             AWS::CloudFormation::Stack      sam-app
-
SS
DELETE_IN_PROGRESS                           AWS::CloudFormation::Stack
  AwsSamAutoDependencyLayerNestedSt         -
                                              ack
DELETE_COMPLETE                             AWS::CloudFormation::Stack
  AwsSamAutoDependencyLayerNestedSt         -
                                              ack
UPDATE_COMPLETE                             AWS::CloudFormation::Stack      sam-app
-

```

CloudFormation outputs from deployed stack

Outputs

```

Key          HelloWorldFunctionIamRole
Description   Implicit IAM Role created for Hello World function
Value        arn:aws:iam::012345678910:role/sam-app-
HelloWorldFunctionRole-15GL0UR9LMT1W

Key          HelloWorldApi
Description   API Gateway endpoint URL for Prod stage for Hello World
function
Value        https://<restapiid>.execute-api.us-west-2.amazonaws.com/Prod/
hello/

Key          HelloWorldFunction
Description   Hello World Lambda Function ARN
Value        arn:aws:lambda:us-west-2:012345678910:function:sam-app-
HelloWorldFunction-yQDNe17r9maD

```

Successfully created/updated stack - sam-app in us-west-2

¡Su aplicación ya está implementada y ejecutándose en el Nube de AWS!

Paso 4: ejecutar la aplicación

En este paso, enviarás una solicitud GET al punto de conexión de la API y verás el resultado de la función de Lambda.

Para obtener el valor del punto de conexión de tu API

1. A partir de la información que muestra la AWS SAM CLI en el paso anterior, localiza la sección `Outputs`. En esta sección, localiza tu recurso de `HelloWorldApi` para encontrar el valor del punto de conexión HTTP. El siguiente es un ejemplo de output:

```
-----  
Outputs  
-----
```

```
...
```

```
Key                HelloWorldApi
```

```
Description        API Gateway endpoint URL for Prod stage for Hello World  
function
```

```
Value              https://ets1gv8lxi.execute-api.us-west-2.amazonaws.com/Prod/  
hello/
```

```
...  
-----
```

2. También puedes utilizar el comando `sam list endpoints --output json` para obtener esta información. El siguiente es un ejemplo de output:

```
$ sam list endpoints --output json
```

```
2023-03-15 12:39:19 Loading policies from IAM...
```

```
2023-03-15 12:39:25 Finished loading policies from IAM.
```

```
[  
  {  
    "LogicalResourceId": "HelloWorldFunction",  
    "PhysicalResourceId": "sam-app-HelloWorldFunction-yQDNe17r9maD",  
    "CloudEndpoint": "-",  
    "Methods": "-"  
  },  
  {  
    "LogicalResourceId": "ServerlessRestApi",  
    "PhysicalResourceId": "ets1gv8lxi",  
    "CloudEndpoint": [  
      "https://ets1gv8lxi.execute-api.us-west-2.amazonaws.com/Prod",  
      "https://ets1gv8lxi.execute-api.us-west-2.amazonaws.com/Stage"
```

```
    ],  
    "Methods": [  
        "/hello['get']"  
    ]  
  }  
]
```

Para invocar tu función

- Mediante el navegador o la línea de comandos, envía una solicitud GET al punto de conexión de la API. A continuación, se muestra un ejemplo con el comando curl:

```
$ curl https://ets1gv8lxi.execute-api.us-west-2.amazonaws.com/Prod/hello/  
{"message": "hello world"}
```

Paso 5: Interactúa con tu función en el Nube de AWS

En este paso, utiliza la AWS SAM CLI para invocar la función de Lambda en la Nube de AWS.

Para invocar la función de eventos de Lambda en la nube

1. Toma nota del LogicalResourceId de la función del paso anterior. Debería ser HelloWorldFunction.
2. En tu línea de comandos, desde el directorio del proyecto sam-app, ejecuta lo siguiente:

```
$ sam remote invoke HelloWorldFunction --stack-name sam-app
```

3. AWS SAMCLI invoca su función en la nube y devuelve una respuesta. El siguiente es un ejemplo de output:

```
$ sam remote invoke HelloWorldFunction --stack-name sam-app  
  
Invoking Lambda Function HelloWorldFunction  
START RequestId: d5ef494b-5f45-4086-86fd-d7322fa1a1f9 Version: $LATEST  
END RequestId: d5ef494b-5f45-4086-86fd-d7322fa1a1f9  
REPORT RequestId: d5ef494b-5f45-4086-86fd-d7322fa1a1f9 Duration: 6.62 ms  
Billed Duration: 7 ms Memory Size: 128 MB Max Memory Used: 67 MB Init  
Duration: 164.06 ms  
{"statusCode":200,"body":{"\message\":"hello world\"}}%
```

Paso 6: Modifique y sincronice su aplicación con Nube de AWS

En este paso, utilizará el AWS SAMCLI `sam sync --watch` comando para sincronizar los cambios locales en Nube de AWS.

Para usar la sincronización `sam`

1. En tu línea de comandos, desde el directorio del proyecto `sam-app`, ejecuta lo siguiente:

```
$ sam sync --watch
```

2. La AWS SAM CLI le pide que confirme que está sincronizando una pila de desarrollo. Como el `sam sync --watch` comando implementa automáticamente los cambios locales Nube de AWS en tiempo real, lo recomendamos únicamente para entornos de desarrollo.

La AWS SAM CLI realiza una implementación inicial antes de comenzar a monitorear los cambios locales. El siguiente es un ejemplo de output:

```
$ sam sync --watch
The SAM CLI will use the AWS Lambda, Amazon API Gateway, and AWS StepFunctions APIs
to upload your code without
performing a CloudFormation deployment. This will cause drift in your
CloudFormation stack.
**The sync command should only be used against a development stack**.

Confirm that you are synchronizing a development stack.

Enter Y to proceed with the command, or enter N to cancel:
[Y/n]: y
Queued infra sync. Waiting for in progress code syncs to complete...
Starting infra sync.
Manifest is not changed for (HelloWorldFunction), running incremental build
Building codeuri: /Users/.../Demo/sam-tutorial1/sam-app/hello_world runtime:
python3.9 metadata: {} architecture: x86_64 functions: HelloWorldFunction
Running PythonPipBuilder:CopySource

Build Succeeded

Successfully packaged artifacts and wrote output template to file /var/
folders/45/5ct135bx3fn2551_pt15g6_80000gr/T/tmpq3x9vh63.
Execute the following command to deploy the packaged template
```

```
sam deploy --template-file /var/folders/45/5ct135bx3fn2551_pt15g6_80000gr/T/
tmpq3x9vh63 --stack-name <YOUR STACK NAME>
```

Deploying with following values

=====

```
Stack name           : sam-app
Region               : us-west-2
Disable rollback     : False
Deployment s3 bucket : aws-sam-cli-managed-default-samclisam-s3-demo-
bucket-1a4x26zbcdkqr
Capabilities          : ["CAPABILITY_NAMED_IAM",
"CAPABILITY_AUTO_EXPAND"]
Parameter overrides  : {}
Signing Profiles      : null
```

Initiating deployment

=====

2023-03-15 13:10:05 - Waiting for stack create/update to complete

CloudFormation events from stack operations (refresh every 0.5 seconds)

ResourceStatus	ResourceType	LogicalResourceId	ResourceStatusReason
UPDATE_IN_PROGRESS	AWS::CloudFormation::Stack		Transformation succeeded
CREATE_IN_PROGRESS	AWS::CloudFormation::Stack	AwsSamAutoDependencyLayerNestedSt	-
			ack
CREATE_IN_PROGRESS	AWS::CloudFormation::Stack	AwsSamAutoDependencyLayerNestedSt	Resource creation Initiated
			ack
CREATE_COMPLETE	AWS::CloudFormation::Stack	AwsSamAutoDependencyLayerNestedSt	-
			ack
UPDATE_IN_PROGRESS	AWS::Lambda::Function	HelloWorldFunction	-
UPDATE_COMPLETE	AWS::Lambda::Function	HelloWorldFunction	-
UPDATE_COMPLETE_CLEANUP_IN_PROGRE	AWS::CloudFormation::Stack		
			ack

```

SS
UPDATE_COMPLETE                               AWS::CloudFormation::Stack      sam-app
-----

CloudFormation outputs from deployed stack
-----

Outputs
-----
Key                HelloWorldFunctionIamRole
Description        Implicit IAM Role created for Hello World function
Value              arn:aws:iam::012345678910:role/sam-app-
HelloWorldFunctionRole-15GL0UR9LMT1W

Key                HelloWorldApi
Description        API Gateway endpoint URL for Prod stage for Hello World
function
Value              https://ets1gv8lxi.execute-api.us-west-2.amazonaws.com/Prod/
hello/

Key                HelloWorldFunction
Description        Hello World Lambda Function ARN
Value              arn:aws:lambda:us-west-2:012345678910:function:sam-app-
HelloWorldFunction-yQDNe17r9maD
-----

Stack update succeeded. Sync infra completed.

Infra sync completed.
CodeTrigger not created as CodeUri or DefinitionUri is missing for
ServerlessRestApi.

```

A continuación, modificará el código de tu función de Lambda. AWS SAMCLIDetectará automáticamente este cambio y sincronizará la aplicación con el Nube de AWS.

Para modificar y sincronizar tu aplicación

1. Abre el archivo `sam-app/hello_world/app.py` con el IDE que prefiera.
2. Cambia el message y guarda el archivo. A continuación, se muestra un ejemplo:

```
import json
```

```
...
def lambda_handler(event, context):
    ...
    return {
        "statusCode": 200,
        "body": json.dumps({
            "message": "hello everyone!",
            ...
        }),
    }
}
```

3. AWS SAMCLIDetecta su cambio y sincroniza su aplicación con. Nube de AWS El siguiente es un ejemplo de output:

```
Syncing Lambda Function HelloWorldFunction...
Manifest is not changed for (HelloWorldFunction), running incremental build
Building codeuri: /Users/.../Demo/sam-tutorial1/sam-app/hello_world runtime:
python3.9 metadata: {} architecture: x86_64 functions: HelloWorldFunction
Running PythonPipBuilder:CopySource
Finished syncing Lambda Function HelloWorldFunction.
```

4. Para verificar el cambio, vuelves a enviar una solicitud GET al punto de conexión de la API.

```
$ curl https://ets1gv8lxi.execute-api.us-west-2.amazonaws.com/Prod/hello/
{"message": "hello everyone!"}
```

Paso 7: (opcional) probar tu aplicación localmente

Note

Este paso es opcional.

Important

Este paso requiere que tenga Docker en su equipo local. Debe tener Docker instalado y configurado para utilizar la CLI de AWS SAM para pruebas locales. Para obtener más información, consulte [Instalar Docker](#).

En este paso, se utiliza el AWS SAMCLI `sam local` comando para probar la aplicación de forma local. Para ello, la AWS SAM CLI crea un entorno local utilizando Docker. Este entorno local emula el entorno de ejecución basado en la nube de tu función de Lambda.

Debes hacer lo siguiente:

1. Crea un entorno local para tu función de Lambda e invócala.
2. Aloja tu punto de conexión de la API HTTP localmente y úsalo para invocar tu función de Lambda.

Para invocar la función de eventos de Lambda localmente

1. En tu línea de comandos, desde el directorio del proyecto `sam-app`, ejecuta lo siguiente:

```
$ sam local invoke
```

2. La AWS SAM CLI crea un contenedor de Docker local e invoca tu función. El siguiente es un ejemplo de output:

```
$ sam local invoke
Invoking app.lambda_handler (python3.9)
Local image was not found.
Removing rapid images for repo public.ecr.aws/sam/emulation-python3.9
Building image.....
Using local image: public.ecr.aws/lambda/python:3.9-rapid-x86_64.

Mounting /Users/.../Demo/sam-tutorial1/sam-app/.aws-sam/build/HelloWorldFunction
as /var/task:ro,delegated inside runtime container
START RequestId: b046db01-2a00-415d-af97-35f3a02e9eb6 Version: $LATEST
END RequestId: b046db01-2a00-415d-af97-35f3a02e9eb6
REPORT RequestId: b046db01-2a00-415d-af97-35f3a02e9eb6      Init Duration: 1.01 ms
      Duration: 633.45 ms      Billed Duration: 634 ms      Memory Size: 128 MB      Max
Memory Used: 128 MB
{"statusCode": 200, "body": "{\"message\": \"hello world\"}"}
```

Para alojar tu API de forma local

1. En tu línea de comandos, desde el directorio del proyecto `sam-app`, ejecuta lo siguiente:

```
$ sam local start-api
```

2. La AWS SAM CLI crea un contenedor de Docker local para la función de Lambda y crea un servidor HTTP local para simular el punto de conexión de la API. El siguiente es un ejemplo de output:

```
$ sam local start-api
Initializing the lambda functions containers.
Local image is up-to-date
Using local image: public.ecr.aws/lambda/python:3.9-rapid-x86_64.

Mounting /Users/.../Demo/sam-tutorial1/sam-app/.aws-sam/build/HelloWorldFunction
as /var/task:ro,delegated inside runtime container
Containers Initialization is done.
Mounting HelloWorldFunction at http://127.0.0.1:3000/hello [GET]
You can now browse to the above endpoints to invoke your functions. You do not
need to restart/reload SAM CLI while working on your functions, changes will be
reflected instantly/automatically. If you used sam build before running local
commands, you will need to re-run sam build for the changes to be picked up. You
only need to restart AWS SAM CLI if you update your AWS SAM template
2023-03-15 14:25:21 WARNING: This is a development server. Do not use it in a
production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:3000
2023-03-15 14:25:21 Press CTRL+C to quit
```

3. Mediante el navegador o la línea de comandos, envía una solicitud GET al punto de conexión de la API local. A continuación, se muestra un ejemplo con el comando curl:

```
$ curl http://127.0.0.1:3000/hello
{"message": "hello world"}
```

Paso 8: Elimine la aplicación del Nube de AWS

En este paso, utilizará el AWS SAMCLI `sam delete` comando para eliminar la aplicación del Nube de AWS.

Para eliminar la aplicación del Nube de AWS

1. En tu línea de comandos, desde el directorio del proyecto `sam-app`, ejecuta lo siguiente:

```
$ sam delete
```


2. La AWS SAM CLI le pedirá que confirme. A continuación, eliminará el bucket y la CloudFormation pila de Amazon S3 de la aplicación. El siguiente es un ejemplo de output:

```
$ sam delete
Are you sure you want to delete the stack sam-app in the region us-west-2 ? [y/N]: y
Are you sure you want to delete the folder sam-app in S3 which contains the artifacts? [y/N]: y
- Deleting S3 object with key c6ce8fa8b5a97dd022ecd006536eb5a4
- Deleting S3 object with key 5d513a459d062d644f3b7dd0c8b56a2a.template
- Deleting S3 object with key sam-app/2bebf67c79f6a743cc5312f6dfc1efee.template
- Deleting S3 object with key sam-app/6b208d0e42ad15d1cee77d967834784b.template
- Deleting S3 object with key sam-app/da3c598813f1c2151579b73ad788cac8
- Deleting S3 object with key sam-app/f798cdd93cee188a71d120f14a035b11
- Deleting Cloudformation stack sam-app
```

```
Deleted successfully
```

Resolución de problemas

Para solucionar el problema AWS SAMCLI, consulte [AWS SAM CLI solución de problemas](#).

Más información

Para seguir informándose sobre ellos AWS SAM, consulta los siguientes recursos:

- [El taller completo de AWS SAM](#): un taller diseñado para enseñarle muchas de las principales características que AWS SAM ofrece.
- [Sesiones con SAM](#): serie de vídeos sobre su uso AWS SAM creada por nuestro equipo de AWS Serverless Developer Advocate.
- [Serverless Land](#): sitio que reúne la información más reciente, blogs, vídeos, códigos y recursos de aprendizaje acerca del AWS sin servidor.

Conversión de funciones de eventos de Lambda en AWS SAM Aplicaciones

Si tiene una función en la consola Lambda que desea administrar en VS Code como infraestructura como código (IaC), puede transferirla a VS Code y, a continuación, convertirla en una plantilla. AWS

SAM Con la función convertida en una AWS SAM plantilla, puede controlar el control de versiones de su infraestructura, automatizar las implementaciones, depurar funciones de forma remota y mantener entornos coherentes durante todo el ciclo de vida del desarrollo. Con las funciones de depuración local y remota, también puede solucionar problemas de forma más eficaz revisando el código, inspeccionando las variables y estableciendo puntos de interrupción tanto a nivel local como en la nube. AWS

Para obtener instrucciones sobre cómo mover las funciones de la consola y configurar el desarrollo local, consulte [Desarrollo local de funciones de Lambda con VS Code](#) en la Guía para desarrolladores de AWS Lambda . Siga estas instrucciones para mover una función de evento Lambda de la consola a VS Code y, a continuación, convertirla en una AWS SAM plantilla. Una vez hecho esto, puede realizar lo siguiente:

- Depure su función de forma remota en la AWS nube. Para obtener más información, consulte [Depuración de forma remota de funciones de Lambda con Visual Studio Code](#) en la Guía para desarrolladores de AWS Lambda .
- Administrar los recursos y permisos asociados en la infraestructura y en el código. Para obtener más información, consulte [Uso de Lambda con la infraestructura como código \(IaC\)](#) en la Guía para desarrolladores de AWS Lambda
- Implemente y actualice su aplicación mediante. AWS SAMCLI Para obtener más información, consulte [AWS SAM CLI](#).
- Diseñar visualmente su arquitectura sin servidor con Infrastructure Composer. Para obtener más información, consulte [la Guía para desarrolladores de Infrastructure Composer](#).

Más información

Para seguir aprendiendo sobre AWS SAM ellos, consulte los siguientes recursos:

- [El taller completo de AWS SAM](#): un taller diseñado para enseñarle muchas de las principales características que AWS SAM ofrece.
- [Sesiones con SAM](#): serie de vídeos sobre su uso AWS SAM creada por nuestro equipo de AWS Serverless Developer Advocate.
- [Serverless Land](#): sitio que reúne la información más reciente, blogs, vídeos, códigos y recursos de aprendizaje acerca del AWS sin servidor.

Cómo usar AWS Serverless Application Model (AWS SAM)

Las principales herramientas que utiliza para desarrollar su aplicación son la AWS SAM plantilla AWS SAMCLI y el AWS SAM proyecto (que es el directorio de proyectos de su aplicación). Puede utilizar estas herramientas para lo siguiente:

1. [Desarrolle su la aplicación](#) (esto incluye inicializar la aplicación, definir los recursos y crear la aplicación).
2. [Prueba de la aplicación](#).
3. [Depuración de la aplicación](#).
4. [Implementación de la aplicación y los recursos](#).
5. [Supervisión de la aplicación](#).

AWS SAM crea el AWS SAM proyecto después de ejecutar el `sam init` comando y completar su flujo de trabajo posterior. La aplicación sin servidor se define añadiendo código al AWS SAM proyecto. Si bien el AWS SAM proyecto consiste en un conjunto de archivos y carpetas, el archivo más importante es la AWS SAM plantilla (denominada `template.yaml`). En esta plantilla, escribe el código para expresar los recursos, las asignaciones de orígenes de eventos y otras propiedades que definen la aplicación sin servidor.

AWS SAMCLIContiene un repositorio de los comandos que utilizas en tu AWS SAM proyecto. Más específicamente, AWS SAMCLI es lo que usas para compilar, transformar, implementar, depurar, empaquetar, inicializar y sincronizar tu AWS SAM proyecto. En otras palabras, es lo que utilizas para convertir tu AWS SAM proyecto en una aplicación sin servidor.

Para obtener más detalles sobre estas herramientas, consulte los siguientes capítulos:

[AWS SAM CLI](#): obtenga información sobre la CLI de AWS SAM y cómo utilizarla para compilar y ejecutar sus aplicaciones sin servidor.

[AWS SAM proyecto y AWS SAM plantilla](#): aprenda sobre la estructura del AWS SAM proyecto y cómo usar la AWS SAM plantilla para definir su aplicación sin servidor.

AWS SAM CLI

AWS Serverless Application Model La interfaz de línea de comandos (AWS SAMCLI) es una herramienta de línea de comandos para el desarrollo y las pruebas locales de aplicaciones sin servidor. AWS SAMCLILe permite crear, transformar, implementar, depurar, empaquetar, inicializar y sincronizar sus aplicaciones sin servidor de forma local antes de implementarlas en la nube.

La CLI de AWS SAM funciona con aplicaciones sin servidor que se definen mediante diferentes marcos y herramientas de infraestructura como código (IaC), con distintos niveles de soporte:

- AWS SAM plantillas: proporciona soporte nativo con todo el conjunto de funciones, incluidas las capacidades locales de prueba, depuración, empaquetado e implementación.
- AWS Aplicaciones CDK: admite pruebas locales de funciones de eventos de Lambda después de sintetizar la aplicación CloudFormation CDK en plantillas mediante AWS el comando `cdk synth`.
- CloudFormation plantillas: ofrece compatibilidad directa porque AWS SAM se extiende CloudFormation y admite los recursos sin servidor que se definen en las plantillas estándar. CloudFormation
- Aplicaciones Terraform: brindan soporte limitado para la creación y las pruebas locales de funciones de Lambda. Requiere que genere artefactos de AWS SAM plantilla que representen las funciones de Terraform-defined Lambda.

Para obtener el soporte de funciones más completo y una experiencia de desarrollador optimizada, le recomendamos que utilice plantillas nativas AWS SAM .

Temas

- [Cómo AWS SAM CLI los comandos están documentados](#)
- [Configuración de la AWS SAM de CLI](#)
- [AWS SAM CLI Comandos principales de la de](#)
- [Pruebas locales con la de AWS SAM CLI](#)

Cómo AWS SAM CLI los comandos están documentados

Los comandos CLI de AWS SAM se documentan con el siguiente formato:

- **Solicitud:** la solicitud de Linux está documentada de forma predeterminada y se muestra como (\$). En los comandos específicos de Windows, (>) se utiliza como símbolo del sistema. No incluya el símbolo al escribir comandos.
- **Directorio:** cuando los comandos se deben ejecutar desde un directorio específico, el nombre del directorio se muestra antes del símbolo de comando.
- **Entrada del usuario:** el texto del comando que se debe ingresar en la línea de comando se formatea como **user input**.
- **Texto reemplazable:** el texto variable, como los nombres de los archivos y los parámetros, tiene el formato siguiente *replaceable text*. En comandos de varias líneas o comandos en los que se requiere una entrada específica desde el teclado, la entrada de teclado también se puede mostrar como texto reemplazable. Por ejemplo, *ENTER*.
- **Salida:** la salida devuelta como respuesta al comando tiene el formato de `computer output`.

El siguiente comando `sam deploy` y la salida son un ejemplo:

```
$ sam deploy --guided --template template.yaml
```

```
Configuring SAM deploy
```

```
=====
```

```
Looking for config file [samconfig.toml] : Found
```

```
Reading default arguments : Success
```

```
Setting default arguments for 'sam deploy'
```

```
=====
```

```
Stack Name [sam-app]: ENTER
```

```
AWS Region [us-west-2]: ENTER
```

```
#Shows you resources changes to be deployed and require a 'Y' to initiate deploy
```

```
Confirm changes before deploy [y/N]: ENTER
```

```
#SAM needs permission to be able to create roles to connect to the resources in  
your template
```

```
Allow SAM CLI IAM role creation [Y/n]: ENTER
```

```
#Preserves the state of previously provisioned resources when an operation fails
```

```
Disable rollback [y/N]: ENTER
```

```
HelloWorldFunction may not have authorization defined, Is this okay? [y/N]: y
```

```
Save arguments to configuration file [Y/n]: ENTER
```

```
SAM configuration file [samconfig.toml]: ENTER
```

```
SAM configuration environment [default]: ENTER
```

1. `sam deploy --guided --template template.yaml` es el comando que se introduce en la línea de comandos.
2. **`sam deploy --guided --template`** debes proporcionarse tal cual.
3. `template.yaml` se puede reemplazar por el nombre de archivo específico.
4. La salida comienza en Configuring SAM deploy.
5. En el resultado, *y* indique **ENTER** los valores reemplazables que proporcione.

Configuración de la AWS SAM de CLI

Una de las ventajas AWS SAM es que optimiza el tiempo del desarrollador al eliminar las tareas repetitivas. AWS SAMCLI incluye un archivo de configuración denominado `asísamconfig`. De forma predeterminada, no AWS SAMCLI es necesaria ninguna configuración, pero puede actualizar el archivo de configuración para poder ejecutar comandos con menos parámetros y, en su lugar, hacer referencia AWS SAM a los parámetros personalizados en el archivo de configuración. En los ejemplos de la siguiente tabla se muestra cómo se pueden optimizar los comandos:

Original	Optimización con samconfig
<code>sam build --cached --parallel --use-containers</code>	<code>sam build</code>
<code>sam local invoke --env-vars locals.json</code>	<code>sam local invoke</code>
<code>sam local start-api --env-vars locals.json --warm-containers EAGER</code>	<code>sam local start-api</code>

La CLI de AWS SAM proporciona un conjunto de comandos para ayudar a los desarrolladores a crear, desarrollar e implementar aplicaciones sin servidor. Cada uno de estos comandos se puede configurar con indicadores opcionales según las preferencias de la aplicación y del desarrollador. Para obtener más información, consulte el [AWS SAMCLI contenido de GitHub](#)

En los temas de esta sección se indica cómo crear su [Archivo de configuración de la CLI de AWS SAM](#) y personalizar su configuración predeterminada para optimizar el tiempo de desarrollo de su aplicación sin servidor.

Temas

- [Cómo crear el archivo de configuración \(el archivo samconfig\)](#)

- [Configurar los ajustes del proyecto](#)
- [Configura las credenciales y los ajustes básicos](#)

Cómo crear el archivo de configuración (el archivo **samconfig**)

El archivo AWS SAMCLI de configuración (nombre de archivosamconfig) es un archivo de texto que normalmente utiliza la estructura TOML, pero también puede estar en YAML. Cuando se utiliza una plantilla de inicio AWS rápido, este archivo se crea al ejecutar el `sam init` comando. Puede actualizar este archivo al implementar una aplicación mediante el comando `sam deploy -i -g`.

Una vez completada la implementación, el archivo `samconfig` contiene un perfil con el nombre `default` si utilizó los valores predeterminados. Al volver a ejecutar el `deploy` comando, AWS SAM se aplican las opciones de configuración almacenadas en este perfil.

La ventaja del `samconfig` archivo es que AWS SAM almacena los ajustes de configuración de cualquier otro comando disponible además del comando `deploy`. Además de estos valores creados en una nueva implementación, hay una serie de atributos que puedes configurar en el `samconfig` archivo para simplificar otros aspectos del flujo de trabajo del desarrollador AWS SAMCLI.

Configurar los ajustes del proyecto

Puede especificar los ajustes específicos del proyecto, como los valores de los parámetros de AWS SAMCLI comando, en un archivo de configuración para usarlo con el. AWS SAMCLI Para obtener más información acerca de este archivo de configuración, consulta [Archivo de configuración de la CLI de AWS SAM](#).

Uso de archivos de configuración

Los archivos de configuración se estructuran por entorno, comando y valor de parámetro. Para obtener más información, consulta [Fundamentos de los archivos de configuración](#).

Para configurar un nuevo entorno

1. Especifica el entorno que se va a utilizar en el archivo de configuración de .

El siguiente es un ejemplo de cómo especificar un entorno `prod` nuevo:

TOML

```
[prod.global.parameters]
```

YAML

```
prod:
  global:
    parameters:
```

2. Especifica los valores de los parámetros como pares clave-valor en la sección de parámetros del archivo de configuración.

A continuación, se muestra un ejemplo de cómo especificar el nombre de la pila de la aplicación para el entorno prod.

TOML

```
[prod.global.parameters]
stack_name = "prod-app"
```

YAML

```
prod:
  global:
    parameters:
      stack_name: prod-app
```

3. Utiliza la opción `--config-env` para especificar el entorno que se va a utilizar.

A continuación, se muestra un ejemplo:

```
$ sam deploy --config-env "prod"
```

Para configurar los valores de parámetros

1. Especifica el comando de la CLI de AWS SAM para el que desea configurar los valores de los parámetros. Para configurar los valores de los parámetros de todos los comandos de la CLI de AWS SAM, utiliza el identificador `global`.

El siguiente es un ejemplo de cómo especificar los valores de los parámetros para el comando `sam deploy` del entorno `default`:

TOML

```
[default.deploy.parameters]
confirm_changeset = true
```

YAML

```
default:
  deploy:
    parameters:
      confirm_changeset: true
```

A continuación, se muestra un ejemplo de cómo especificar los valores de los parámetros para todos los comandos de la CLI de AWS SAM del entorno `default`:

TOML

```
[default.global.parameters]
stack_name = "sam-app"
```

YAML

```
default:
  global:
    parameters:
      stack_name: sam-app
```

2. También puede especificar los valores de los parámetros y modificar el archivo de configuración mediante el flujo interactivo de la CLI de AWS SAM.

A continuación, se muestra un ejemplo de flujo interactivo con `sam deploy --guided`.

```
$ sam deploy --guided
```

```
Configuring SAM deploy
=====
```

```

Looking for config file [samconfig.toml] : Found
Reading default arguments : Success

Setting default arguments for 'sam deploy'
=====
Stack Name [sam-app]: ENTER
AWS Region [us-west-2]: ENTER
#Shows you resources changes to be deployed and require a 'Y' to initiate
deploy
Confirm changes before deploy [Y/n]: n
#SAM needs permission to be able to create roles to connect to the resources in
your template
Allow SAM CLI IAM role creation [Y/n]: ENTER
#Preserves the state of previously provisioned resources when an operation
fails
Disable rollback [y/N]: ENTER
HelloWorldFunction may not have authorization defined, Is this okay? [y/N]: y
Save arguments to configuration file [Y/n]: ENTER
SAM configuration file [samconfig.toml]: ENTER
SAM configuration environment [default]: ENTER

```

Para obtener más información, consulta [Crear y modificar archivos de configuración](#).

Ejemplos

Ejemplo básico TOML

A continuación, se muestra un ejemplo de un archivo de configuración de `samconfig.toml`:

```

...
version = 0.1

[default]
[default.global]
[default.global.parameters]
stack_name = "sam-app"

[default.build.parameters]
cached = true
parallel = true

[default.deploy.parameters]

```

```
capabilities = "CAPABILITY_IAM"
confirm_changeset = true
resolve_s3 = true

[default.sync.parameters]
watch = true

[default.local_start_api.parameters]
warm_containers = "EAGER"

[prod]
[prod.sync]
[prod.sync.parameters]
watch = false
```

Ejemplo básico YAML

A continuación, se muestra un ejemplo de un archivo de configuración de `samconfig.yaml`:

```
version 0.1
default:
  global:
    parameters:
      stack_name: sam-app
  build:
    parameters:
      cached: true
      parallel: true
  deploy:
    parameters:
      capabilities: CAPABILITY_IAM
      confirm_changeset: true
      resolve_s3: true
  sync:
    parameters:
      watch: true
  local_start_api:
    parameters:
      warm_containers: EAGER
prod:
  sync:
    parameters:
      watch: false
```

Configura las credenciales y los ajustes básicos

Utilice el AWS Command Line Interface (AWS CLI) para configurar los ajustes básicos, como AWS las credenciales, el nombre de la región por defecto y el formato de salida por defecto. Una vez configurados, puede usar estos ajustes con la CLI de AWS SAM. Para obtener más información, consulta lo siguiente en la Guía del usuario de AWS Command Line Interface :

- [Fundamentos de configuración](#)
- [Opciones de los archivos de configuración y credenciales](#)
- [Perfiles con nombre para AWS CLI](#)
- [Uso de un perfil con nombre habilitado para IAM Identity Center](#)

Para obtener instrucciones de Quick Setup, consulta [Paso 2: utilice el AWS CLI para configurar AWS credentials](#).

AWS SAM CLI Comandos principales de la de

La CLI de AWS SAM tiene algunos comandos básicos que puede utilizar para crear, compilar, probar, implementar y sincronizar su aplicación sin servidor. En la siguiente tabla se enumeran estos comandos y se proporcionan enlaces con más información sobre cada uno de ellos.

Para obtener una lista completa de los comandos de la CLI de AWS SAM, consulte [Referencia de los comandos AWS SAM CLI](#).

Comando	¿Qué hace?	Temas relacionados
sam build	Prepara una aplicación para los pasos posteriores del flujo de trabajo del desarrollador, como las pruebas locales o la implementación AWS en la nube.	<ul style="list-style-type: none">• Introducción a la construcción con AWS SAM• compilación de sam
sam deploy	Implementa una aplicación en la AWS nube mediante CloudFormation.	<ul style="list-style-type: none">• Introducción a la implementación con AWS SAM• sam deploy

Comando	¿Qué hace?	Temas relacionados
sam init	Proporciona opciones para inicializar y crear una nueva aplicación sin servidor.	<ul style="list-style-type: none"> • Crea tu aplicación en AWS SAM • sam init
sam local	Proporciona los subcomandos para probar las aplicaciones sin servidor de forma local.	<ul style="list-style-type: none"> • Introducción a las pruebas con el sam local comando • sam local generate-event • invocar sam local • sam local start-api • sam local start-lambda
sam remote invoke	Proporciona una forma de interactuar con AWS los recursos compatibles en la AWS nube.	<ul style="list-style-type: none"> • Introducción a las pruebas en la nube con invocación remota de sam • invocación remota de sam
sam remote test-event	Proporciona una forma de acceder a los eventos de prueba que se pueden compartir para las funciones de eventos de AWS Lambda y gestionarlos.	<ul style="list-style-type: none"> • Introducción a las pruebas en la nube con evento de prueba remoto de sam • evento de prueba remoto de sam
sam sync	Ofrece opciones para sincronizar rápidamente los cambios en las aplicaciones locales con la AWS nube.	<ul style="list-style-type: none"> • Introducción al uso sam sync sincronizar con Nube de AWS • sam sync

Pruebas locales con la de AWS SAM CLI

La CLI de AWS SAM permite probar las aplicaciones sin servidor de forma local en diferentes herramientas de infraestructura como código (IaC). Esta guía explica cómo utilizar la CLI de AWS SAM para realizar pruebas locales con varios marcos de IaC.

Los beneficios principales de utilizar la CLI de AWS SAM para las pruebas locales son:

- Desarrollo rápido: pruebe los cambios de código sin desplegarlos en AWS
- Rentabilidad: desarrolle y pruebe sin incurrir en gastos AWS
- Capacidad sin conexión: trabaje en sus aplicaciones sin conexión a Internet
- Depuración simplificada: revise el código de la función de eventos Lambda de forma local mediante un depurador
- Pruebas realistas: pruebe sus aplicaciones mediante la emulación local de servicios AWS

Test AWS SAM aplicaciones a nivel local

Para obtener información sobre cómo probar aplicaciones definidas mediante AWS SAM plantillas, consulte [Probar y depurar aplicaciones sin servidor](#) en esta guía.

Test CloudFormation plantillas de forma local

Para utilizar las funciones AWS SAMCLI de prueba locales de la CloudFormation aplicación, añada la AWS SAM transformación a tu CloudFormation plantilla. Para obtener más información, consulte la [anatomía de la plantilla de AWS SAM](#) para realizar lo siguiente:

- Añadir una AWS SAM transformación a las CloudFormation plantillas
- Comprender la compatibilidad de las plantillas
- Explorar la sintaxis de los recursos sin servidor

Test AWS Aplicaciones CDK a nivel local

Puede usarlo AWS SAMCLI para probar las aplicaciones AWS CDK localmente después de sintetizar la aplicación CDK en CloudFormation plantillas mediante el comando `AWS cdk synth`.

Para obtener más información, consulte [Pruebe y cree aplicaciones de CDK de forma local con la guía para desarrolladores de AWS CDK](#). AWS SAMCLI AWS Esto incluye:

- [Introducción a las pruebas locales](#)
- [Prueba local de aplicaciones de AWS CDK con AWS SAM](#)

Probar las aplicaciones de Terraform de forma local

La CLI de AWS SAM es compatible con los proyectos de Terraform y Terraform Cloud. Puede utilizarla para realizar la depuración y pruebas locales de las funciones y capas de Lambda y las API de HTTP y REST de Amazon API Gateway.

Para configurar el entorno y obtener información sobre todas las funciones disponibles, consulte [Compatibilidad con Terraform](#) en esta guía. Esto incluye:

- [Introducción a compatibilidad de Terraform para la CLI de AWS SAM](#)
- [Uso de la AWS SAM de CLI con Terraform para la depuración y las pruebas locales](#)

AWS SAM plantilla

Tras ejecutar el `sam init` comando y completar su flujo de trabajo posterior, AWS SAM crea el directorio del proyecto de la aplicación, que es su AWS SAM proyecto. Para definir la aplicación sin servidor, añada código a su AWS SAM proyecto. Si bien su AWS SAM proyecto consta de un conjunto de archivos y carpetas, el archivo con el que trabaja principalmente es su AWS SAM plantilla (denominada `template.yaml`). En esta plantilla, se escribe el código para expresar los recursos, las asignaciones de orígenes de eventos y otras propiedades que definen la aplicación sin servidor.

Note

Un elemento clave de la AWS SAM plantilla es la especificación AWS SAM de la plantilla. Esta especificación proporciona una sintaxis abreviada que, en comparación con ella CloudFormation, permite utilizar menos líneas de código para definir los recursos, las asignaciones de fuentes de eventos, los permisos, las API y otras propiedades de la aplicación sin servidor.

En esta sección se proporcionan detalles sobre cómo utilizar las secciones de la AWS SAM plantilla para definir los tipos de recursos, las propiedades de los recursos, los tipos de datos, los atributos de los recursos, las funciones intrínsecas y las extensiones de API Gateway.

AWS SAM las plantillas son una extensión de CloudFormation las plantillas, con tipos de sintaxis únicos que utilizan una sintaxis abreviada con menos líneas de código que. CloudFormation Esto acelera el desarrollo al crear una aplicación sin servidor. Para obtener más información, consulta [AWS SAM recursos y propiedades](#). Para obtener la referencia completa de CloudFormation las plantillas, consulte la [Referencia CloudFormation de plantillas](#) en la Guía del AWS CloudFormation usuario.

Al desarrollar, a menudo te resultará beneficioso dividir el código de la aplicación en archivos separados para organizar y administrar mejor la aplicación. Un ejemplo básico de esto es usar un archivo independiente para el código de la función de AWS Lambda evento en lugar de tener este código en la AWS SAM plantilla. Para ello, organice el código de la función Lambda en un subdirectorio de su proyecto y haga referencia a su ruta local dentro de la plantilla (`.`). AWS Serverless Application Model AWS SAM

Temas

- [AWS SAM anatomía de la plantilla](#)
- [AWS SAM recursos y propiedades](#)
- [Generado CloudFormation recursos para AWS SAM](#)
- [Atributos de recursos compatibles con AWS SAM](#)
- [Extensiones de API Gateway para AWS SAM](#)
- [Funciones intrínsecas para AWS SAM](#)
- [CloudFormation compatibilidad con extensiones de idioma](#)

AWS SAM anatomía de la plantilla

Un archivo AWS SAM de plantilla sigue de cerca el formato de un archivo de CloudFormation plantilla, que se describe en [Anatomía de la plantilla](#) de la Guía del AWS CloudFormation usuario. Las principales diferencias entre los archivos AWS SAM de CloudFormation plantilla y los archivos de plantilla son las siguientes:

- Declaración de transformación. La declaración Transform: `AWS::Serverless-2016-10-31` es obligatoria para los archivos de plantilla de AWS SAM. Esta declaración identifica un archivo CloudFormation de plantilla como un archivo AWS SAM de plantilla. Para obtener más información, consulta la [Transformación](#) en la AWS CloudFormation Guía del usuario de .
- Sección Globals. La Globals sección es exclusiva de AWS SAM. Define las propiedades que son comunes a todas las funciones sin servidor y APIs. Todos los `AWS::Serverless::StateMachine` recursos `AWS::Serverless::Function`, `AWS::Serverless::Api`, `AWS::Serverless::CapacityProvider`, `AWS::Serverless::HttpApi`, `AWS::Serverless::SimpleTable`, y heredan las propiedades que se definen en la Globals sección. Para obtener más información, consulta [La sección Globals del AWS SAM plantilla](#).
- Sección de recursos. En AWS SAM las plantillas, la Resources sección puede contener una combinación de CloudFormation recursos y AWS SAM recursos. Para obtener más información sobre CloudFormation los recursos, consulte la [referencia sobre los tipos de AWS recursos y propiedades](#) en la Guía del AWS CloudFormation usuario. Para obtener más información sobre AWS SAM los recursos, consulte [AWS SAM recursos y propiedades](#).

Todas las demás secciones de un archivo de AWS SAM plantilla corresponden a la sección del archivo de CloudFormation plantilla del mismo nombre.

YAML

El siguiente ejemplo muestra un fragmento de código de plantilla con formato YAML.

```
Transform: AWS::Serverless-2016-10-31
```

```
Globals:
```

```
  set of globals
```

```
Description:
```

```
  String
```

```
Metadata:
```

```
  template metadata
```

```
Parameters:
```

```
  set of parameters
```

```
Mappings:
```

```
  set of mappings
```

```
Conditions:
```

```
  set of conditions
```

```
Resources:
```

```
  set of resources
```

```
Outputs:
```

```
  set of outputs
```

Secciones de la plantilla

AWS SAM las plantillas pueden incluir varias secciones principales. Solo se requieren las secciones Transform y Resources.

Puedes incluir las secciones de la plantilla en cualquier orden. Sin embargo, si usa extensiones de idioma, debe agregar `AWS::LanguageExtensions` antes de la transformación sin servidor (es decir, antes de `AWS::Serverless-2016-10-31`), como se muestra en el siguiente ejemplo:

```
Transform:
```

```
  - AWS::LanguageExtensions
```

- AWS::Serverless-2016-10-31

Al crear la plantilla, puede ser útil utilizar el orden lógico de la siguiente lista. Esto se debe a que los valores de una sección pueden hacer referencia a valores de una sección anterior.

Transformar (obligatorio)

En el AWS SAM caso de las plantillas, debe incluir esta sección con un valor de `AWS::Serverless-2016-10-31`.

Las transformaciones adicionales son opcionales. Para obtener más información, consulta la [Transformación](#) en la AWS CloudFormation Guía del usuario de .

Globals (opcional)

Propiedades comunes a todas las funciones sin servidor y a APIs las tablas sencillas. Todos los `AWS::Serverless::StateMachine` recursos `AWS::Serverless::Function`, `AWS::Serverless::Api`, `AWS::Serverless::CapacityProvider`, `AWS::Serverless::HttpApi`, `AWS::Serverless::SimpleTable`, y heredan las propiedades que se definen en la `Globals` sección.

Esta sección es exclusiva de AWS SAM. No hay una sección correspondiente en las plantillas de CloudFormation .

Descripción (opcional)

Una cadena de texto que describe la plantilla.

Esta sección se corresponde directamente con la `Description` sección de CloudFormation plantillas.

Metadatos (opcional)

Objetos que proporcionan información adicional acerca de la plantilla.

Esta sección se corresponde directamente con la `Metadata` sección de CloudFormation plantillas.

Parámetros (opcional)

Valores que se van a pasar a tu plantilla en tiempo de ejecución (al crear o actualizar una pila). Puedes consultar los parámetros en las secciones `Resources` y `Outputs` de la plantilla. Los objetos que se declaran en la sección `Parameters` hacen que el comando `sam deploy --guided` presente solicitudes adicionales al usuario.

Los valores que se transfieren mediante el parámetro `--parameter-overrides` del comando `sam deploy` (y las entradas del archivo de configuración) tienen prioridad sobre las entradas del archivo de plantilla de AWS SAM. Para obtener más información sobre el comando `sam deploy`, consulta [sam deploy](#) en la referencia de comandos de la AWS SAM CLI. Para obtener más información sobre el archivo de configuración, consulta [Archivo de configuración de la CLI de AWS SAM](#).

[Mapeos \(opcional\)](#)

Un mapeo de claves y valores asociados que puede utilizar para especificar valores de parámetros condicionales, similar a una tabla de búsqueda. Puede asociar una clave a un valor correspondiente mediante la función intrínseca `Fn::FindInMap` en las secciones `Resources` y `Outputs`.

Esta sección se corresponde directamente con la `Mappings` sección de CloudFormation plantillas.

[Condiciones \(opcional\)](#)

Condiciones que controlan si determinados recursos se crean o si a determinadas propiedades de recursos se les asigna un valor durante la creación de la pila o la actualización. Por ejemplo, puede crear un recurso condicionalmente que depende de si la pila es para un entorno de prueba o de producción.

Esta sección se corresponde directamente con la `Conditions` sección de CloudFormation plantillas.

[Recursos \(obligatorio\)](#)

Los recursos de la pila y sus propiedades, como una instancia de Amazon Elastic Compute Cloud (Amazon EC2) o un depósito de Amazon Simple Storage Service (Amazon S3). Puedes consultar los recursos en las secciones `Resources` y `Outputs` de la plantilla.

Esta sección es similar a la sección `Resources` de plantillas de CloudFormation. En AWS SAM las plantillas, esta sección puede contener AWS SAM recursos además de los CloudFormation recursos.

[Salidas \(opcional\)](#)

Los valores que se devuelven siempre que ve las propiedades de su pila. Por ejemplo, puede declarar un resultado para el nombre de un bucket de S3 y, a continuación, llamar al comando `aws cloudformation describe-stacks` AWS Command Line Interface (AWS CLI) para ver el nombre.

Esta sección se corresponde directamente con la sección Outputs de plantillas de CloudFormation .

Pasos a seguir a continuación

Para descargar e implementar un ejemplo de aplicación sin servidor que contenga un archivo de AWS SAM plantilla, consulta [Empezar con AWS SAM](#) y sigue las instrucciones que se indican en [Tutorial: Implemente una aplicación Hello World con AWS SAM](#).

La sección Globals del AWS SAM plantilla

A veces, los recursos que declaras en una AWS SAM plantilla tienen configuraciones comunes. Por ejemplo, es posible que tenga una aplicación con varios recursos de `AWS::Serverless::Function` que tengan configuraciones Runtime, Memory, VPCConfig, Environment y Cors idénticas. En lugar de duplicar esta información en todos los recursos, puede declararlos una vez en la sección Globals y dejar que los recursos los hereden.

La Globals sección admite los siguientes tipos AWS SAM de recursos:

- `AWS::Serverless::Api`
- `AWS::Serverless::CapacityProvider`
- `AWS::Serverless::Function`
- `AWS::Serverless::HttpApi`
- `AWS::Serverless::SimpleTable`
- `AWS::Serverless::StateMachine`
- `AWS::Serverless::MicrovmImage`
- `AWS::Serverless::NetworkConnector`

Ejemplo:

```
Globals:
  Function:
    Runtime: nodejs12.x
    Timeout: 180
    Handler: index.handler
    Environment:
      Variables:
```

```
TABLE_NAME: data-table

Resources:
  HelloWorldFunction:
    Type: AWS::Serverless::Function
    Properties:
      Environment:
        Variables:
          MESSAGE: "Hello From SAM"

  ThumbnailFunction:
    Type: AWS::Serverless::Function
    Properties:
      Events:
        Thumbnail:
          Type: Api
          Properties:
            Path: /thumbnail
            Method: POST
```

En este ejemplo, tanto HelloWorldFunction como ThumbnailFunction utilizan “nodejs12.x” para Runtime, “180” segundos para Timeout e “index.handler” para Handler. HelloWorldFunction agrega la variable de entorno MESSAGE, además del TABLE_NAME heredado. ThumbnailFunction hereda todas las propiedades de Globals y añade un origen de eventos de la API.

Recursos y propiedades compatibles

AWS SAM admite los siguientes recursos y propiedades.

```
Globals:
  Api:
    AccessLogSetting:
    Auth:
    BinaryMediaTypes:
    CacheClusterEnabled:
    CacheClusterSize:
    CanarySetting:
    Cors:
    DefinitionUri:
    Domain:
    EndpointConfiguration:
    EndpointAccessMode:
```

```
GatewayResponses:
MethodSettings:
MinimumCompressionSize:
Name:
OpenApiVersion:
PropagateTags:
SecurityPolicy:
TracingEnabled:
Variables:
```

```
CapacityProvider:
  InstanceRequirements:
  KmsKeyArn:
  LoggingConfig:
  ManagedResourceTags:
  OperatorRole:
  PropagateTags:
  ScalingConfig:
  Tags:
  VpcConfig:
```

```
Function:
  Architectures:
  AssumeRolePolicyDocument:
  AutoPublishAlias:
  AutoPublishAliasAllProperties:
  CapacityProviderConfig:
  CodeSigningConfigArn:
  CodeUri:
  DeadLetterQueue:
  DeploymentPreference:
  Description:
  DurableConfig:
  Environment:
  EphemeralStorage:
  EventInvokeConfig:
  FileSystemConfigs:
  FunctionScalingConfig:
  FunctionUrlConfig:
  Handler:
  KmsKeyArn:
  Layers:
  LoggingConfig:
  MemorySize:
```

```
PermissionsBoundary:
PropagateTags:
ProvisionedConcurrencyConfig:
PublishToLatestPublished:
RecursiveLoop:
ReservedConcurrentExecutions:
RolePath:
Runtime:
RuntimeManagementConfig:
SnapStart:
SourceKMSKeyArn:
Tags:
TenancyConfig:
Timeout:
Tracing:
VersionDeletionPolicy:
VpcConfig:
```

```
HttpApi:
  AccessLogSettings:
  Auth:
  PropagateTags:
  StageVariables:
  Tags:
```

```
SimpleTable:
  SSESpecification:
```

```
StateMachine:
  PropagateTags:
```

```
MicrovmImage:
  BaseImageArn:
  BaseImageVersion:
  BuildRoleArn:
  Hooks:
  EnvironmentVariables:
  EgressNetworkConnectors:
  CpuConfigurations:
  Resources:
  AdditionalOsCapabilities:
  Logging:
  Tags:
  PropagateTags:
```



```
NetworkConnector:
  OperatorRole:
  Tags:
  PropagateTags:
```

Note

No se admiten recursos y propiedades que no se incluyan en la lista anterior. Algunas de las razones por las que no se admiten son: 1) Presentan posibles problemas de seguridad o 2) Hacen que la plantilla sea difícil de entender.

APIs implícitas

AWS SAM crea API implícitas cuando declaras una API en la `Events` sección. Se puede utilizar `Globals` para anular todas las propiedades de las API implícitas.

Propiedades anulables

Los recursos pueden anular las propiedades que declare en la sección `Globals`. Por ejemplo, puede añadir nuevas variables a un mapa de variables de entorno o puede anular las variables declaradas globalmente. Sin embargo, el recurso no puede eliminar una propiedad especificada en la sección `Globals`.

De forma más general, la sección `Globals` declara las propiedades que comparten todos los recursos. Algunos recursos pueden proporcionar nuevos valores para las propiedades declaradas globalmente, pero no pueden eliminarlos. Si algunos recursos usan una propiedad pero otros no, no debe declararlos en la sección `Globals`.

En las secciones siguientes, se describe cómo funciona la anulación para distintos tipos de datos.

Se sustituyen los tipos de datos primitivos

Entre estos tipos de datos primitivos se incluyen cadenas, números, valores booleanos, etc.

El valor especificado en la sección `Resources` reemplaza al valor de la sección `Globals`.

Ejemplo:

```
Globals:
```

```
Function:
  Runtime: nodejs12.x

Resources:
  MyFunction:
    Type: AWS::Serverless::Function
    Properties:
      Runtime: python3.9
```

El Runtime para MyFunction está establecido en python3.9.

Los mapas se fusionan

Los mapas también se conocen como diccionarios o colecciones de pares de clave-valor.

Las entradas de mapas de la sección Resources se fusionan con las entradas de mapas globales. Si hay duplicados, la entrada de la sección Resource anula la entrada de la sección Globals.

Ejemplo:

```
Globals:
  Function:
    Environment:
      Variables:
        STAGE: Production
        TABLE_NAME: global-table

Resources:
  MyFunction:
    Type: AWS::Serverless::Function
    Properties:
      Environment:
        Variables:
          TABLE_NAME: resource-table
          NEW_VAR: hello
```

Las variables de entorno MyFunction se establecen de la siguiente manera:

```
{
  "STAGE": "Production",
  "TABLE_NAME": "resource-table",
  "NEW_VAR": "hello"
```

```
}
```

Las listas son acumulativas

Las listas también se conocen como matrices.

Las entradas de la lista de la sección `Globals` se anteponen a la lista de la sección `Resources`.

Ejemplo:

```
Globals:
  Function:
    VpcConfig:
      SecurityGroupIds:
        - sg-123
        - sg-456

Resources:
  MyFunction:
    Type: AWS::Serverless::Function
    Properties:
      VpcConfig:
        SecurityGroupIds:
          - sg-first
```

Los `SecurityGroupIds` para la `VpcConfig` de `MyFunction` establecen de la siguiente manera:

```
[ "sg-123", "sg-456", "sg-first" ]
```

AWS SAM recursos y propiedades

En esta sección se describen los tipos de recursos y propiedades específicos de AWS SAM. Estos recursos y propiedades se definen mediante la sintaxis AWS SAM abreviada. AWS SAM también admite los tipos CloudFormation de recursos y propiedades. Para obtener información de referencia sobre todos los tipos de AWS recursos y propiedades CloudFormation y el AWS SAM soporte, consulte la [referencia sobre los tipos de AWS recursos y propiedades](#) en la Guía del AWS CloudFormation usuario.

Temas

- [AWS::Serverless: :API](#)
- [AWS::Serverless::Application](#)
- [AWS::Serverless::CapacityProvider](#)
- [AWS::Serverless: :Conector](#)
- [AWS::Serverless: :Función](#)
- [AWS::Serverless:: API GraphQL](#)
- [AWS::Serverless::HttpApi](#)
- [AWS::Serverless::WebSocketApi](#)
- [AWS::Serverless::LayerVersion](#)
- [AWS::Serverless::MicrovmImage](#)
- [AWS::Serverless::NetworkConnector](#)
- [AWS::Serverless::SimpleTable](#)
- [AWS::Serverless::StateMachine](#)
- [AWS::Serverless::WebFunction](#)

AWS::Serverless: :API

Crea una colección de recursos y métodos de Amazon API Gateway que se pueden invocarse a través de puntos de conexión HTTPS.

No es necesario añadir un [AWS::Serverless: :API](#) recurso de forma explícita a una plantilla de definición de aplicaciones AWS sin servidor. Un recurso de este tipo se crea implícitamente a partir de la unión de eventos de API definidos en los recursos [AWS::Serverless: :Función](#) definidos en la plantilla que no hacen referencia a un recurso [AWS::Serverless: :API](#).

Se debe utilizar un [AWS::Serverless: :API](#) recurso para definir y documentar la API que se utiliza OpenApi, lo que proporciona más capacidad para configurar los recursos subyacentes de Amazon API Gateway.

Te recomendamos que utilices CloudFormation enlaces o políticas de IAM para comprobar que los recursos de API Gateway tienen autorizadores adjuntos para controlar el acceso a ellos.

Para obtener más información sobre el uso de CloudFormation ganchos, consulte [Registrar ganchos](#) en la guía del usuario de la CloudFormation CLI y en el repositorio [GitHub apigw-enforce-authorizer](#).

Para obtener más información sobre el uso de las políticas de IAM, consulta [Exigir que las rutas de API tengan autorización](#) en la Guía para desarrolladores de API Gateway.

Note

Cuando despliegas en, transforma tus recursos en AWS CloudFormation recursos AWS SAM . AWS SAM CloudFormation Para obtener más información, consulte [Generado CloudFormation recursos para AWS SAM](#).

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
Type: AWS::Serverless::Api
Properties:
  AccessLogSetting: AccessLogSetting
  AlwaysDeploy: Boolean
  ApiKeySourceType: String
  Auth: ApiAuth
  BinaryMediaTypes: List
  CacheClusterEnabled: Boolean
  CacheClusterSize: String
  CanarySetting: CanarySetting
  Cors: String | CorsConfiguration
  DefinitionBody: JSON
  DefinitionUri: String | ApiDefinition
  Description: String
  DisableExecuteApiEndpoint: Boolean
  Domain: DomainConfiguration
  EndpointAccessMode: String
  EndpointConfiguration: EndpointConfiguration
  FailOnWarnings: Boolean
  GatewayResponses: Map
  MergeDefinitions: Boolean
  MethodSettings: MethodSettings
  MinimumCompressionSize: Integer
  Mode: String
  Models: Map
```

```
Name: String  
OpenApiVersion: String  
PropagateTags: Boolean  
Policy: JSON  
SecurityPolicy: String  
StageName: String  
Tags: Map  
TracingEnabled: Boolean  
Variables: Map
```

Propiedades

AccessLogSetting

Configuración de registros de acceso para una etapa.

Tipo: [AccessLogSetting](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [AccessLogSetting](#) propiedad de un `AWS::ApiGateway::Stage` recurso.

AlwaysDeploy

Siempre implementas la API, incluso cuando no se han detectado cambios en la API.

Tipo: Booleano

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

ApiKeySourceType

Origen de la clave de API para medir las solicitudes de acuerdo con un plan de uso. Los valores válidos son `HEADER` y `AUTHORIZER`.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [ApiKeySourceType](#) propiedad de un `AWS::ApiGateway::RestApi` recurso.

Auth

Configura la autorización para controlar el acceso a su API de API Gateway.

Para obtener más información sobre cómo configurar el acceso mediante, AWS SAM consulte [Controle el acceso a la API con su AWS SAM plantilla](#). Para ver un ejemplo que muestra cómo anular un autorizador global, consulte [the section called “Anule un autorizador global para su API de REST de Amazon API Gateway”](#).

Tipo: [ApiAuth](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

BinaryMediaTypes

Lista de tipos MIME que su API puede devolver. Utilice esto para habilitar la compatibilidad con datos binarios para las API.

Tipo: lista

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es similar a la [BinaryMediaTypes](#) propiedad de un `AWS::ApiGateway::RestApi` recurso. La lista de BinaryMediaTypes se añade tanto al CloudFormation recurso como al documento de OpenAPI.

CacheClusterEnabled

Indica si el almacenado en caché está habilitado para la etapa. Para almacenar en caché las respuestas, también debe configurar `CachingEnabled` para `true` bajo `MethodSettings`.

Tipo: Booleano

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [CacheClusterEnabled](#) propiedad de un `AWS::ApiGateway::Stage` recurso.

CacheClusterSize

El tamaño del clúster de caché de la etapa.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [CacheClusterSize](#) propiedad de un `AWS::ApiGateway::Stage` recurso.

CanarySetting

Configure una prueba canario como una etapa de una implementación normal.

Tipo: [CanarySetting](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [CanarySetting](#) propiedad de un `AWS::ApiGateway::Stage` recurso.

Cors

Gestione el uso compartido de Cross-origin recursos (CORS) para todas sus API de API Gateway. Especifica el dominio que quieres permitir como cadena o especifica un diccionario con una configuración de Cors adicional.

Note

CORS requiere modificar AWS SAM la definición de OpenAPI. Cree una definición de OpenAPI en línea en `DefinitionBody` para activar CORS.

Para obtener más información sobre CORS, consulta [Habilitar CORS para un recurso de la API de REST de API Gateway](#) en la Guía para desarrolladores de API Gateway.

Tipo: Cadena | [CorsConfiguration](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

DefinitionBody

Especificación de OpenAPI que describe su API. Si no se especifica ni `DefinitionUri` ni `DefinitionBody`, SAM generará `DefinitionBody` para usted en función de la configuración de la plantilla.

Para hacer referencia a un archivo local de OpenAPI que defina su API, usa la transformación `AWS::Include`. Para obtener más información, consulta [¿Cómo AWS SAM carga los archivos locales](#).

Tipo: JSON

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es similar a la [Body](#) propiedad de un `AWS::ApiGateway::RestApi` recurso. Si se proporcionan determinadas propiedades, el contenido se puede insertar o modificar en ellas `DefinitionBody` antes de pasarlo a ellas CloudFormation. Las propiedades incluyen `Auth`, `BinaryMediaTypes`, `Cors`, `GatewayResponses`, `Models` y una `EventSource` del tipo `Api` para una `AWS::Serverless::Function` correspondiente.

DefinitionUri

Amazon S3 Uri, la ruta de archivo local o el objeto de ubicación del documento de OpenAPI que define la API. El objeto Amazon S3 al que hace referencia esta propiedad debe ser un archivo OpenAPI válido. Si no se especifica ni `DefinitionUri` ni `DefinitionBody`, SAM generará `DefinitionBody` para usted en función de la configuración de la plantilla.

Si se proporciona una ruta de archivo local, la plantilla debe seguir el flujo de trabajo que incluye el comando `sam deploy` o `sam package` para que la definición se transforme correctamente.

Las funciones intrínsecas no se admiten en OpenApi los archivos externos a los que hace referencia `DefinitionUri`. En su lugar, utilice la `DefinitionBody` propiedad con la [transformación Include](#) para importar una OpenApi definición a la plantilla.

Tipo: Cadena | [ApiDefinition](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es similar a la [BodyS3Location](#) propiedad de un `AWS::ApiGateway::RestApi` recurso. Las propiedades anidadas de Amazon S3 tienen un nombre diferente.

Description

Una descripción del recurso de la Api.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Description](#) propiedad de un `AWS::ApiGateway::RestApi` recurso.

DisableExecuteApiEndpoint

Especifica si los clientes pueden invocar la API mediante el punto de conexión `execute-api` predeterminado. De forma predeterminada, los clientes pueden invocar su API con el `https://{api_id}.execute-api.{region}.amazonaws.com` predeterminado. Para exigir que los clientes utilicen un nombre de dominio personalizado para invocar su API, especifica `True`.

Tipo: Booleano

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es similar a la [DisableExecuteApiEndpoint](#) propiedad de un `AWS::ApiGateway::RestApi` recurso. Se pasa directamente a la propiedad `disableExecuteApiEndpoint` de una extensión [x-amazon-apigateway-endpoint-configuration](#), que se agrega a la propiedad [Body](#) de un recurso `AWS::ApiGateway::RestApi`.

Domain

Configura un dominio personalizado para esta API de API Gateway.

Tipo: [DomainConfiguration](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

EndpointConfiguration

El tipo de punto de conexión de una API de REST.

Tipo: [EndpointConfiguration](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es similar a la [EndpointConfiguration](#) propiedad de un `AWS::ApiGateway::RestApi` recurso. Las propiedades anidadas de configuración tienen un nombre diferente.

EndpointAccessMode

El modo de acceso al punto final de la API REST. Se requiere cuando se utilizan políticas de seguridad mejoradas (las que tienen el prefijo `SecurityPolicy_`).

Valores válidos: STRICT o BASIC

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [EndpointAccessMode](#) propiedad de un `AWS::ApiGateway::RestApi` recurso.

FailOnWarnings

Especifica si se va a restaurar la creación de API (`true`) o no (`false`) cuando se encuentra una advertencia. El valor predeterminado es `false`.

Tipo: Booleano

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [FailOnWarnings](#) propiedad de un `AWS::ApiGateway::RestApi` recurso.

GatewayResponses

Configura las respuestas de Gateway para una API. Las respuestas de Gateway son respuestas devueltas por API Gateway, ya sea directamente o mediante el uso de autorizadores Lambda. Para obtener más información, consulte la documentación de la [OpenApi extensión Api Gateway para Gateway Responses](#).

Tipo: mapa

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

MergeDefinitions

AWS SAM genera una OpenAPI especificación a partir de la fuente de eventos de la API. Especifique `true` si desea AWS SAM fusionarla con la OpenAPI especificación en línea definida en su `AWS::Serverless::Api` recurso. Especifica que `false` no se fusione.

MergeDefinitions requiere que la propiedad de `DefinitionBody` para `AWS::Serverless::Api` se defina. MergeDefinitions no es compatible con la propiedad de `DefinitionUri` para `AWS::Serverless::Api`.

Valor predeterminado: `false`

Tipo: Booleano

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

MethodSettings

Configura todos los ajustes de la etapa de la API, incluidos los de registro, métricas, `cacheTTL` y la limitación controlada.

Tipo: lista de [MethodSetting](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [MethodSettings](#) propiedad de un `AWS::ApiGateway::Stage` recurso.

MinimumCompressionSize

Permite la compresión de los cuerpos de respuesta en función del `Accept-Encoding` encabezado del cliente. La compresión se desencadena cuando el tamaño del cuerpo de respuesta es superior o igual al umbral configurado. El umbral de tamaño máximo del cuerpo es de 10 MB (10.485.760 bytes). - Se admiten los siguientes tipos de compresión: `gzip`, `deflate` e `identidad`.

Tipo: entero

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [MinimumCompressionSize](#) propiedad de un `AWS::ApiGateway::RestApi` recurso.

Mode

Esta propiedad solo se aplica cuando se utiliza OpenAPI para definir su API de REST. El Mode determina cómo API Gateway maneja las actualizaciones de recursos. Para obtener más información, consulta la propiedad [Modo](#) en el tipo de recurso [AWS::ApiGateway::RestApi](#).

Valores válidos: `overwrite` o `merge`

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Mode](#) propiedad de un `AWS::ApiGateway::RestApi` recurso.

Models

Los esquemas que deben utilizar sus métodos API. Estos esquemas se pueden describir mediante JSON o YAML. Consulte la sección de ejemplos al final de esta página para ver ejemplos de modelos.

Tipo: mapa

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

Name

Un nombre para el RestApi recurso API Gateway

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Name](#) propiedad de un `AWS::ApiGateway::RestApi` recurso.

OpenApiVersion

Versión de OpenApi a utilizar. Puede ser 2.0 para la especificación Swagger o para una de las versiones OpenApi 3.0, por ejemplo. 3.0.1 Para más información sobre OpenAPI, consulta la [Especificación de OpenAPI](#).

Note

AWS SAM crea un escenario llamado Stage por defecto. Si se establece esta propiedad en cualquier valor válido, se impedirá la creación de la etapa Stage.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

PropagateTags

Indique si quiere pasar o no las etiquetas de la propiedad Tags a los recursos generados [AWS::Serverless::Api](#). Especifica True para propagar las etiquetas en los recursos generados.

Tipo: Booleano

Obligatorio: no

Valor predeterminado: False

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

Policy

Documento de la política que contiene los permisos para la API. Para establecer el ARN de la política, utilice la función intrínseca !Join con "" como delimitador y los valores "execute-api:/" y "*".

Tipo: JSON

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la propiedad [Policy](#) de un `AWS::ApiGateway::RestApi` recurso.

SecurityPolicy

La versión TLS más el conjunto de cifrado para la API REST. Cuando utilice políticas de seguridad mejoradas (las que tienen el prefijo `SecurityPolicy`), también debe configurar `EndpointAccessMode`.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [SecurityPolicy](#) propiedad de un `AWS::ApiGateway::RestApi` recurso.

StageName

El nombre de la etapa, que API Gateway utiliza como el primer segmento de la ruta en el Identificador uniforme de recursos (URI) invocado.

Para hacer referencia al recurso de la etapa, utilice `<api-logical-id>.Stage`. Para obtener más información sobre cómo hacer referencia a los recursos generados cuando se especifica un recurso [AWS::Serverless::API](#), consulta [CloudFormation recursos generados cuando AWS::Serverless::Api se especifica](#). Para obtener información general sobre CloudFormation los recursos generados, consulte [Generado CloudFormation recursos para AWS SAM](#).

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad es similar a la [StageName](#) propiedad de un `AWS::ApiGateway::Stage` recurso. Es obligatorio en SAM, pero no en API Gateway.

Notas adicionales: la API implícita tiene el nombre de etapa de “Prod”.

Tags

Un mapa (cadena a cadena) que especifica las etiquetas que deben añadirse a esta etapa de la API Gateway. Para obtener más información sobre las claves y los valores válidos de las etiquetas, consulta la [etiqueta de recursos](#) en la Guía del usuario de AWS CloudFormation.

Tipo: mapa

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es similar a la [Tags](#) propiedad de un `AWS::ApiGateway::Stage` recurso. La propiedad Tags de SAM se compone de Key:Value pares; en CloudFormation ella se compone de una lista de objetos Tag.

TracingEnabled

Indica si el rastreo activo con X-Ray está activado para el escenario. Para obtener más información X-Ray, consulte [Rastrear las solicitudes de los usuarios a las API de REST mediante X-Ray](#) la Guía para desarrolladores de API Gateway.

Tipo: Booleano

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [TracingEnabled](#) propiedad de un `AWS::ApiGateway::Stage` recurso.

Variables

Un mapa (cadena a cadena) que define las variables de la fase, donde el nombre de la variable es la clave y el valor de la variable es el valor. Los nombres de variables están limitados a caracteres alfanuméricos. Los valores deben coincidir con las siguientes expresiones regulares: `[A-Za-z0-9._~:/?#&=, - ]+`.

Tipo: mapa

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Variables](#) propiedad de un `AWS::ApiGateway::Stage` recurso.

valores devueltos

Ref.

Cuando se proporciona el ID lógico de este recurso a la función intrínseca Ref, devuelve el ID del API de API Gateway subyacente.

Para obtener más información sobre el uso de la función Ref, consulta [Ref](#) en la Guía del usuario de AWS CloudFormation .

Ventilador:: GetAtt

`Fn::GetAtt` devuelve un valor para un atributo especificado de este tipo. A continuación se indican los atributos disponibles y los valores devueltos de muestra.

Para obtener más información acerca de cómo usar `Fn::GetAtt`, consulta [Fn::GetAtt](#) en la Guía del usuario de AWS CloudFormation .

RootResourceId

El ID del recurso raíz de un recurso `RestApi`, como por ejemplo `a0bc123d4e`.

Ejemplos

SimpleApiExample

Un archivo de AWS SAM plantilla de Hello World que contiene una función de eventos de Lambda con un punto final de API. Se trata de un archivo de AWS SAM plantilla completo para una aplicación sin servidor que funcione.

YAML

```
AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31
Description: AWS SAM template with a simple API definition
Resources:
  ApiGatewayApi:
    Type: AWS::Serverless::Api
    Properties:
      StageName: prod
  ApiFunction: # Adds a GET method at the root resource via an Api event
    Type: AWS::Serverless::Function
    Properties:
      Events:
        ApiEvent:
          Type: Api
          Properties:
            Path: /
            Method: get
            RestApiId:
              Ref: ApiGatewayApi
      Runtime: python3.10
```

```
Handler: index.handler
InlineCode: |
  def handler(event, context):
    return {'body': 'Hello World!', 'statusCode': 200}
```

ApiCorsExample

Un fragmento AWS SAM de plantilla con una API definida en un archivo Swagger externo junto con integraciones de Lambda y configuraciones de CORS. Esto es solo una parte de un archivo de plantilla que muestra una definición. AWS SAM [AWS::Serverless::API](#)

YAML

```
Resources:
  ApiGatewayApi:
    Type: AWS::Serverless::Api
    Properties:
      StageName: Prod
      # Allows www.example.com to call these APIs
      # SAM will automatically add AllowMethods with a list of methods for this API
      Cors: "'www.example.com'"
      DefinitionBody: # Pull in an OpenApi definition from S3
        'Fn::Transform':
          Name: 'AWS::Include'
          # Replace "bucket" with your bucket name
          Parameters:
            Location: s3://bucket/swagger.yaml
```

ApiCognitoAuthExample

Un fragmento AWS SAM de plantilla con una API que utiliza Amazon Cognito para autorizar las solicitudes contra la API. Es solo una parte de un archivo de AWS SAM plantilla que muestra una definición. [AWS::Serverless::API](#)

YAML

```
Resources:
  ApiGatewayApi:
    Type: AWS::Serverless::Api
    Properties:
      StageName: Prod
```

```
Cors: ""
Auth:
  DefaultAuthorizer: MyCognitoAuthorizer
  Authorizers:
    MyCognitoAuthorizer:
      UserPoolArn:
        Fn::GetAtt: [MyCognitoUserPool, Arn]
```

ApiModelsExample

Un fragmento AWS SAM de plantilla con una API que incluye un esquema de modelos. Esto es solo una parte de un archivo de AWS SAM plantilla, que muestra una [AWS::Serverless::API](#) definición con dos esquemas de modelo.

YAML

```
Resources:
  ApiGatewayApi:
    Type: AWS::Serverless::Api
    Properties:
      StageName: Prod
      Models:
        User:
          type: object
          required:
            - username
            - employee_id
          properties:
            username:
              type: string
            employee_id:
              type: integer
            department:
              type: string
        Item:
          type: object
          properties:
            count:
              type: integer
            category:
              type: string
            price:
              type: integer
```

Ejemplo de almacenamiento en caché

Un archivo de AWS SAM plantilla de Hello World que contiene una función Lambda con un punto final de API. La API tiene habilitado el almacenamiento en caché para un recurso y un método. Para obtener más información acerca del almacenamiento en caché, consulta [Habilitar el almacenamiento en caché de la API para mejorar la capacidad de respuesta](#) en la Guía para desarrolladores de API Gateway.

YAML

```
AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31
Description: AWS SAM template with a simple API definition with caching turned on
Resources:
  ApiGatewayApi:
    Type: AWS::Serverless::Api
    Properties:
      StageName: prod
      CacheClusterEnabled: true
      CacheClusterSize: '0.5'
      MethodSettings:
        - ResourcePath: /
          HttpMethod: GET
          CachingEnabled: true
          CacheTtlInSeconds: 300
      Tags:
        CacheMethods: All

  ApiFunction: # Adds a GET method at the root resource via an Api event
    Type: AWS::Serverless::Function
    Properties:
      Events:
        ApiEvent:
          Type: Api
          Properties:
            Path: /
            Method: get
            RestApiId:
              Ref: ApiGatewayApi
      Runtime: python3.10
      Handler: index.handler
      InlineCode: |
        def handler(event, context):
```

```
return {'body': 'Hello World!', 'statusCode': 200}
```

Ejemplo de dominio personalizado con una API privada

Un archivo de AWS SAM plantilla de Hello World que contiene una función Lambda con un punto final de API asignado a un dominio privado. La plantilla crea una asociación de acceso a un dominio entre un punto de conexión de VPC y un dominio privado. Para obtener más información, consulte [Nombres de dominio personalizados para API privadas en API Gateway](#).

YAML

```
AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31
Description: AWS SAM template configured with a custom domain using a private API

Parameters:
  DomainName:
    Type: String
    Default: mydomain.example.com
  CertificateArn:
    Type: String
  HostedZoneId:
    Type: String
  VpcEndpointId:
    Type: String
  VpcEndpointDomainName:
    Type: String
  VpcEndpointHostedZoneId:
    Type: String

Resources:
  MyFunction:
    Type: AWS::Serverless::Function
    Properties:
      InlineCode: |
        def handler(event, context):
            return {'body': 'Hello World!', 'statusCode': 200}
      Handler: index.handler
      Runtime: python3.13
      Events:
        Fetch:
          Type: Api
          Properties:
```

```

    RestApiId:
      Ref: MyApi
    Method: Get
    Path: /get
MyApi:
  Type: AWS::Serverless::Api
  Properties:
    StageName: Prod
    EndpointConfiguration:
      Type: PRIVATE
      VPCEndpointIds:
        - !Ref VpcEndpointId
  Policy:
    Version: '2012-10-17'
    Statement:
      - Effect: Allow
        Principal: '*'
        Action: execute-api:Invoke
        Resource: execute-api/*
      - Effect: Deny
        Principal: '*'
        Action: execute-api:Invoke
        Resource: execute-api/*
    Condition:
      StringNotEquals:
        aws:SourceVpce: !Ref VpcEndpointId
  Domain:
    DomainName: !Ref DomainName
    CertificateArn: !Ref CertificateArn
    EndpointConfiguration: PRIVATE
    BasePath:
      - /
  Route53:
    HostedZoneId: !Ref HostedZoneId
    VpcEndpointDomainName: !Ref VpcEndpointDomainName
    VpcEndpointHostedZoneId: !Ref VpcEndpointHostedZoneId
  AccessAssociation:
    VpcEndpointId: !Ref VpcEndpointId
  Policy:
    Version: '2012-10-17'
    Statement:
      - Effect: Allow
        Principal: '*'
        Action: execute-api:Invoke

```

```

    Resource: execute-api/*
  - Effect: Deny
    Principal: '*'
    Action: execute-api:Invoke
    Resource: execute-api/*
    Condition:
      StringNotEquals:
        aws:SourceVpce: !Ref VpcEndpointId

```

ApiAuth

Configura la autorización para controlar el acceso a su API de API Gateway.

Para obtener más información y ejemplos de configuración del acceso mediante, AWS SAM consulte [Controle el acceso a la API con su AWS SAM plantilla](#).

Sintaxis

Para declarar esta entidad en su plantilla AWS Serverless Application Model (AWS SAM), utilice la siguiente sintaxis.

YAML

```

AddApiKeyRequiredToCorsPreflight: Boolean
AddDefaultAuthorizerToCorsPreflight: Boolean
ApiKeyRequired: Boolean
Authorizers: CognitoAuthorizer | LambdaTokenAuthorizer | LambdaRequestAuthorizer |
AWS_IAM
DefaultAuthorizer: String
InvokeRole: String
ResourcePolicy: ResourcePolicyStatement
UsagePlan: ApiUsagePlan

```

Note

La propiedad `Authorizers` incluye `AWS_IAM`, pero no se necesita ninguna configuración adicional para `AWS_IAM`. Para ver un ejemplo, consulta [AWS YO SOY](#).

Propiedades

AddApiKeyRequiredToCorsPreflight

Si las propiedades `ApiKeyRequired` y `Cors` están configuradas, la configuración `AddApiKeyRequiredToCorsPreflight` provocará que la clave de API se añada a la propiedad `Options`.

Tipo: Booleano

Obligatorio: no

Valor predeterminado: `True`

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

AddDefaultAuthorizerToCorsPreflight

Si las propiedades `DefaultAuthorizer` y `Cors` están configuradas, la configuración `AddDefaultAuthorizerToCorsPreflight` provocará que se añada el autorizador predeterminado a la propiedad `Options` en la sección `OpenAPI`.

Tipo: Booleano

Obligatorio: no

Valor predeterminado: `True`

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

ApiKeyRequired

Si se establece en `true`, una clave de API es necesaria para todos los eventos de API. Para obtener más información sobre claves de API, consulta el tema [Creación y utilización de planes de uso de claves API](#) en la Guía para desarrolladores de API Gateway.

Tipo: Booleano

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

Authorizers

El autorizador utilizado para controlar el acceso a la API de API Gateway.

Para obtener más información, consulte [Controle el acceso a la API con su AWS SAM plantilla](#).

Tipo: [CognitoAuthorizer](#) | [LambdaTokenAuthorizer](#) | [LambdaRequestAuthorizer](#) | AWS_IAM

Obligatorio: no

Valor predeterminado: Ninguno

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

Notas adicionales: SAM añade los autorizadores a la OpenApi definición de API.

DefaultAuthorizer

Especifica un autorizador predeterminado para una API de API Gateway, que se utilizará para autorizar las llamadas a la API de forma predeterminada.

Note

Si la API EventSource de la función asociada a esta API está configurada para usar permisos de IAM, esta propiedad debe estar establecida en; de lo contrario AWS_IAM, se producirá un error.

Tipo: cadena

Obligatorio: no

Valor predeterminado: Ninguno

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

InvokeRole

Establezca las credenciales de integración para todos los recursos y métodos en este valor.

`CALLER_CREDENTIALS` se asigna a `arn:aws:iam::<user>/`, que utiliza las credenciales de la persona que llama para invocar el punto de conexión.

Valores válidos: `CALLER_CREDENTIALS`, `NONE`, `IAMRoleArn`

Tipo: cadena

Obligatorio: no

Valor predeterminado: `CALLER_CREDENTIALS`

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

ResourcePolicy

Configura la política de recursos para todos los métodos y rutas de una API.

Tipo: [ResourcePolicyStatement](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

Notas adicionales: Esta configuración también se puede definir de forma individual con `AWS::Serverless::Function` mediante [ApiFunctionAuth](#). Esto es obligatorio para APIs con `EndpointConfiguration: PRIVATE`.

UsagePlan

Configura un plan de uso asociado a esta API. Para obtener más información sobre claves de API, consulta el tema [Creación y utilización de planes de uso de claves API](#) en la Guía para desarrolladores de API Gateway.

Esta AWS SAM propiedad genera tres CloudFormation recursos adicionales cuando se establece: an [AWS::ApiGateway::UsagePlan](#), an [AWS::ApiGateway::UsagePlanKey](#), an y an [AWS::ApiGateway::ApiKey](#). Para obtener información acerca de esta situación, consulta [UsagePlan se especifica la propiedad](#). Para obtener información general sobre CloudFormation los recursos generados, consulte [Generado CloudFormation recursos para AWS SAM](#).

Tipo: [ApiUsagePlan](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

Ejemplos

CognitoAuth

Ejemplo de autorización de Cognito

YAML

```
Auth:
  Authorizers:
    MyCognitoAuth:
      UserPoolArn:
        Fn::GetAtt:
          - MyUserPool
          - Arn
      AuthType: "COGNITO_USER_POOLS"
  DefaultAuthorizer: MyCognitoAuth
  InvokeRole: CALLER_CREDENTIALS
  AddDefaultAuthorizerToCorsPreflight: false
  ApiKeyRequired: false
  ResourcePolicy:
    CustomStatements: [{
      "Effect": "Allow",
      "Principal": "*",
      "Action": "execute-api:Invoke",
      "Resource": "execute-api:/Prod/GET/pets",
      "Condition": {
        "IpAddress": {
          "aws:SourceIp": "1.2.3.4"
        }
      }
    }]
  IpRangeDenylist:
    - "10.20.30.40"
```

AWS YO SOY

AWS Ejemplo de IAM

YAML

```
Auth:
  Authorizers: AWS_IAM
```

ApiUsagePlan

Configura un plan de uso para una API de API Gateway. Para obtener más información sobre claves de API, consulta el tema [Creación y utilización de planes de uso de claves API](#) en la Guía para desarrolladores de API Gateway.

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
CreateUsagePlan: String
Description: String
Quota: QuotaSettings
Tags: List
Throttle: ThrottleSettings
UsagePlanName: String
```

Propiedades

CreateUsagePlan

Determina cómo se configura este plan de uso. Los valores válidos son PER_API, SHARED y NONE.

PER_API crea los recursos [AWS::ApiGateway::UsagePlan](#), [AWS::ApiGateway::ApiKey](#) y [AWS::ApiGateway::UsagePlanKey](#) específicos de esta API. Estos recursos tienen IDs la lógica de `<api-logical-id>UsagePlan<api-logical-id>ApiKey`, y `<api-logical-id>UsagePlanKey`, respectivamente.

SHARED crea [AWS::ApiGateway::UsagePlan](#) y [AWS::ApiGateway::UsagePlanKey](#) recursos que se comparten en cualquier API que también tenga CreateUsagePlan: SHARED la misma AWS SAM plantilla. [AWS::ApiGateway::ApiKey](#) Estos recursos tienen una lógica IDs de

`ServerlessUsagePlanServerlessApiKey`, y `ServerlessUsagePlanKey`, respectivamente. Si usas esta opción, te recomendamos que añada una configuración adicional para este plan de uso en un solo recurso de API para evitar definiciones conflictivas y un estado incierto.

NONE inhabilita la creación o la asociación de un plan de uso con esta API. Esto solo es necesario si SHARED o PER_API se especifica en [La sección Globals del AWS SAM plantilla](#).

Valores válidos: PER_API, SHARED y NONE

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

Description

Es una descripción del plan de uso.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Description](#) propiedad de un `AWS::ApiGateway::UsagePlan` recurso.

Quota

Configura el número de solicitudes que pueden realizar los usuarios dentro de un determinado intervalo.

Tipo: [QuotaSettings](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Quota](#) propiedad de un `AWS::ApiGateway::UsagePlan` recurso.

Tags

Una matriz de etiquetas arbitrarias (pares de clave-valor) que se asocian al plan de uso.

Esta propiedad usa el [tipo de CloudFormation etiqueta](#).

Tipo: lista

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Tags](#) propiedad de un `AWS::ApiGateway::UsagePlan` recurso.

Throttle

Configura la tasa de solicitudes general (promedio de solicitudes por segundo) y la capacidad de ráfaga.

Tipo: [ThrottleSettings](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Throttle](#) propiedad de un `AWS::ApiGateway::UsagePlan` recurso.

UsagePlanName

Nombre para el plan de uso.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [UsagePlanName](#) propiedad de un `AWS::ApiGateway::UsagePlan` recurso.

Ejemplos

UsagePlan

El siguiente es un ejemplo de plan de uso.

YAML

```
Auth:
  UsagePlan:
```

```
CreateUsagePlan: PER_API
Description: Usage plan for this API
Quota:
  Limit: 500
  Period: MONTH
Throttle:
  BurstLimit: 100
  RateLimit: 50
Tags:
  - Key: TagName
    Value: TagValue
```

CognitoAuthorizer

Define un autorizador de grupo de usuarios de Amazon Cognito.

Para obtener más información y ejemplos, consulta [Controle el acceso a la API con su AWS SAM plantilla](#).

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
AuthorizationScopes: List
Identity: CognitoAuthorizationIdentity
UserPoolArn: String | List
```

Propiedades

AuthorizationScopes

Lista de los ámbitos de autorización de este autorizador.

Tipo: lista

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

Identity

Esta propiedad se puede usar para especificar una solicitud `IdentitySource` entrante de un autorizador.

Tipo: [CognitoAuthorizationIdentity](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

UserPoolArn

Los ARN del grupo de usuarios de Amazon Cognito que se utilizarán para la autorización. Especifique un único ARN como cadena o varios ARNs como una lista para usar varios grupos de usuarios.

Tipo: Cadenas | Lista

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

Ejemplos

CognitoAuth

Ejemplo de autorización de Cognito

YAML

```
Auth:
  Authorizers:
    MyCognitoAuth:
      AuthorizationScopes:
        - scope1
        - scope2
      UserPoolArn:
        Fn::GetAtt:
          - MyCognitoUserPool
```



```
- Arn
Identity:
  Header: MyAuthorizationHeader
  ValidationExpression: myauthvalidationexpression
```

CognitoAuthorizationIdentity

Esta propiedad se puede usar para especificar una IdentitySource solicitud entrante de un autorizador. Para obtener más información, IdentitySource consulte la extensión [ApiGateway Authorizer OpenApi](#).

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
Header: String
ReauthorizeEvery: Integer
ValidationExpression: String
```

Propiedades

Header

Especifique el nombre del encabezado de la autorización en la OpenApi definición.

Tipo: cadena

Obligatorio: no

Predeterminado: Autorización

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

ReauthorizeEvery

El período time-to-live (TTL), en segundos, que especifica cuánto tiempo API Gateway almacena en caché los resultados del autorizador. Si especificas un valor mayor que 0, API Gateway

almacena en caché las respuestas del autorizador. De forma predeterminada, API Gateway establece esta propiedad en 300. El valor máximo es 3600 o 1 hora.

Tipo: entero

Obligatorio: no

Predeterminado: 300

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un equivalente. CloudFormation

ValidationExpression

Especifica una expresión de validación para la identidad de entrada.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

Ejemplos

CognitoAuthIdentity

YAML

```
Identity:
  Header: MyCustomAuthHeader
  ValidationExpression: Bearer.*
  ReauthorizeEvery: 30
```

LambdaRequestAuthorizer

Configure un autorizador Lambda para controlar el acceso a su API con una función de eventos Lambda.

Para obtener más información y ejemplos, consulta [Controle el acceso a la API con su AWS SAM plantilla](#).

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
DisableFunctionDefaultPermissions: Boolean  
FunctionArn: String  
FunctionInvokeRole: String  
FunctionPayloadType: String  
Identity: LambdaRequestAuthorizationIdentity
```

Propiedades

DisableFunctionDefaultPermissions

Especifique si true desea AWS SAM evitar que se cree automáticamente un `AWS::Lambda::Permissions` recurso para aprovisionar permisos entre el `AWS::Serverless::Api` recurso y la función Lambda autorizadora.

Valor predeterminado: false

Tipo: Booleano

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

FunctionArn

Especifica el ARN de la función de Lambda que proporciona la autorización para la API.

Note

AWS SAM creará automáticamente un `AWS::Lambda::Permissions` recurso cuando `FunctionArn` se especifique para `AWS::Serverless::Api`. El recurso `AWS::Lambda::Permissions` proporciona permisos entre la API y la función autorizador de Lambda.

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

FunctionInvokeRole

Añade las credenciales del autorizador a la OpenApi definición del autorizador Lambda.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un equivalente. CloudFormation

FunctionPayloadType

Esta propiedad se puede utilizar para definir el tipo de autorizador Lambda para una API.

Valores válidos: TOKEN o REQUEST

Tipo: cadena

Obligatorio: no

Valor predeterminado: TOKEN

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

Identity

Esta propiedad se puede usar para especificar una solicitud IdentitySource entrante de un autorizador. Esta propiedad solo es necesaria si la propiedad FunctionPayloadType está establecida en REQUEST.

Tipo: [LambdaRequestAuthorizationIdentity](#)

Obligatorio: condicional

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

Ejemplos

LambdaRequestAuth

YAML

```
Authorizers:
  MyLambdaRequestAuth:
    FunctionPayloadType: REQUEST
    FunctionArn:
      Fn::GetAtt:
        - MyAuthFunction
        - Arn
    FunctionInvokeRole:
      Fn::GetAtt:
        - LambdaAuthInvokeRole
        - Arn
    Identity:
      Headers:
        - Authorization1
```

LambdaRequestAuthorizationIdentity

Esta propiedad se puede usar para especificar una IdentitySource solicitud entrante de un autorizador. Para obtener más información, IdentitySource consulte la extensión [ApiGateway Authorizer OpenApi](#).

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
Context: List
Headers: List
QueryStrings: List
ReauthorizeEvery: Integer
StageVariables: List
```

Propiedades

Context

Convierte las cadenas de contexto dadas en las expresiones de mapeo de formato `context.contextString`.

Tipo: lista

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

Headers

Convierte los encabezados en una cadena de expresiones de mapeo de formato `method.request.header.name` separadas por comas.

Tipo: lista

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

QueryString

Convierte las cadenas de consulta dadas en cadenas de expresiones de asignación con un formato `method.request.querystring.queryString` separadas por comas.

Tipo: lista

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

ReauthorizeEvery

El período time-to-live (TTL), en segundos, que especifica cuánto tiempo API Gateway almacena en caché los resultados del autorizador. Si especificas un valor mayor que 0, API Gateway almacena en caché las respuestas del autorizador. De forma predeterminada, API Gateway establece esta propiedad en 300. El valor máximo es 3600 o 1 hora.

Tipo: entero

Obligatorio: no

Predeterminado: 300

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un equivalente. CloudFormation

StageVariables

Convierte las variables de etapa dadas en cadenas de expresiones de asignación con un formato `stageVariables.stageVariable` separadas por comas.

Tipo: lista

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

Ejemplos

LambdaRequestIdentity

YAML

```
Identity:
  QueryStrings:
    - auth
  Headers:
    - Authorization
  StageVariables:
    - VARIABLE
  Context:
    - authcontext
  ReauthorizeEvery: 100
```

LambdaTokenAuthorizer

Configure un autorizador Lambda para controlar el acceso a su API con una función de eventos Lambda.

Para obtener más información y ejemplos, consulta [Controle el acceso a la API con su AWS SAM plantilla](#).

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
DisableFunctionDefaultPermissions: Boolean  
FunctionArn: String  
FunctionInvokeRole: String  
FunctionPayloadType: String  
Identity: LambdaTokenAuthorizationIdentity
```

Propiedades

DisableFunctionDefaultPermissions

Especifique si true desea AWS SAM evitar que se cree automáticamente un `AWS::Lambda::Permissions` recurso para aprovisionar permisos entre el `AWS::Serverless::Api` recurso y la función Lambda autorizadora.

Valor predeterminado: false

Tipo: Booleano

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

FunctionArn

Especifica el ARN de la función de Lambda que proporciona la autorización para la API.

Note

AWS SAM creará automáticamente un `AWS::Lambda::Permissions` recurso cuando `FunctionArn` se especifique para `AWS::Serverless::Api`. El recurso

`AWS::Lambda::Permissions` proporciona permisos entre la API y la función autorizador de Lambda.

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

FunctionInvokeRole

Añade las credenciales del autorizador a la OpenApi definición del autorizador Lambda.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un equivalente. CloudFormation

FunctionPayloadType

Esta propiedad se puede utilizar para definir el tipo de autorizador Lambda para una API.

Valores válidos: TOKEN o REQUEST

Tipo: cadena

Obligatorio: no

Valor predeterminado: TOKEN

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

Identity

Esta propiedad se puede usar para especificar una solicitud IdentitySource entrante de un autorizador. Esta propiedad solo es necesaria si la propiedad FunctionPayloadType está establecida en REQUEST.

Tipo: [LambdaTokenAuthorizationIdentity](#)

Obligatorio: condicional

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

Ejemplos

LambdaTokenAuth

YAML

```
Authorizers:
  MyLambdaTokenAuth:
    FunctionArn:
      Fn::GetAtt:
        - MyAuthFunction
        - Arn
    Identity:
      Header: MyCustomAuthHeader # OPTIONAL; Default: 'Authorization'
      ValidationExpression: mycustomauthexpression # OPTIONAL
      ReauthorizeEvery: 20 # OPTIONAL; Service Default: 300
```

BasicLambdaTokenAuth

YAML

```
Authorizers:
  MyLambdaTokenAuth:
    FunctionArn:
      Fn::GetAtt:
        - MyAuthFunction
        - Arn
```

LambdaTokenAuthorizationIdentity

Esta propiedad se puede usar para especificar una IdentitySource solicitud entrante de un autorizador. Para obtener más información, IdentitySource consulte la extensión [ApiGateway Authorizer OpenApi](#).

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
Header: String  
ReauthorizeEvery: Integer  
ValidationExpression: String
```

Propiedades

Header

Especifique el nombre del encabezado de la autorización en la OpenApi definición.

Tipo: cadena

Obligatorio: no

Predeterminado: Autorización

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

ReauthorizeEvery

El período time-to-live (TTL), en segundos, que especifica cuánto tiempo API Gateway almacena en caché los resultados del autorizador. Si especificas un valor mayor que 0, API Gateway almacena en caché las respuestas del autorizador. De forma predeterminada, API Gateway establece esta propiedad en 300. El valor máximo es 3600 o 1 hora.

Tipo: entero

Obligatorio: no

Predeterminado: 300

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un equivalente. CloudFormation

ValidationExpression

Especifica una expresión de validación para validar la identidad entrante.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

Ejemplos

LambdaTokenIdentity

YAML

```
Identity:
  Header: MyCustomAuthHeader
  ValidationExpression: Bearer.*
  ReauthorizeEvery: 30
```

ResourcePolicyStatement

Configura una política de recursos para todos los métodos y rutas de una API. Para obtener más información sobre las políticas de recursos, consulta [Controlar el acceso a una API con las políticas de recursos de API Gateway](#) en la Guía para desarrolladores de API Gateway.

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
AwsAccountBlacklist: List
AwsAccountWhitelist: List
CustomStatements: List
IntrinsicVpcBlacklist: List
IntrinsicVpcWhitelist: List
IntrinsicVpceBlacklist: List
IntrinsicVpceWhitelist: List
IpRangeBlacklist: List
IpRangeWhitelist: List
SourceVpcBlacklist: List
SourceVpcWhitelist: List
```

Propiedades

AwsAccountBlacklist

Las AWS cuentas que se van a bloquear.

Tipo: lista de cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

AwsAccountWhitelist

Las AWS cuentas que se van a permitir. Para obtener un ejemplo del uso de esta propiedad, consulta la sección Ejemplos al final de esta página.

Tipo: lista de cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

CustomStatements

Una lista de instrucciones de política de recursos personalizadas para aplicarlas a esta API. Para obtener un ejemplo del uso de esta propiedad, consulta la sección Ejemplos al final de esta página.

Tipo: lista

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

IntrinsicVpcBlacklist

La lista de nubes privadas virtuales (VPCs) que se van a bloquear, en la que cada VPC se especifica como referencia, por ejemplo, una [referencia dinámica](#) o la función Ref [intrínseca](#). Para obtener un ejemplo del uso de esta propiedad, consulta la sección Ejemplos al final de esta página.

Tipo: lista

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

IntrinsicVpcWhitelist

La lista de VPCs permitidos, donde cada VPC se especifica como una referencia, como una [referencia dinámica](#) o la función Ref [intrínseca](#).

Tipo: lista

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

IntrinsicVpceBlacklist

La lista de puntos de conexión de VPC que se van a bloquear, donde cada punto de conexión de VPC se especifica como referencia, por ejemplo, una [referencia dinámica](#) o la [función intrínseca Ref](#).

Tipo: lista

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

IntrinsicVpceWhitelist

La lista de nubes privadas virtuales (VPC) permitidas, donde cada punto de conexión de VPC se especifica como referencia, por ejemplo, una [referencia dinámica](#) o la [función intrínseca Ref](#). Para obtener un ejemplo del uso de esta propiedad, consulta la sección Ejemplos al final de esta página.

Tipo: lista

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

IpRangeBlacklist

Las direcciones IP o los rangos de direcciones que se van a bloquear. Para obtener un ejemplo del uso de esta propiedad, consulta la sección Ejemplos al final de esta página.

Tipo: lista

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

IpRangeWhitelist

Las direcciones IP o los rangos de direcciones que se van a permitir.

Tipo: lista

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

SourceVpcBlacklist

La VPC de origen o los puntos de conexión de VPC que se van a bloquear. Los nombres de la VPC de origen deben empezar por "vpc-" y los nombres de los puntos de conexión de VPC de origen deben empezar por "vpce-". Para obtener un ejemplo del uso de esta propiedad, consulta la sección Ejemplos al final de esta página.

Tipo: lista

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

SourceVpcWhitelist

La VPC de origen o los puntos de conexión de VPC que se van a permitir. Los nombres de la VPC de origen deben empezar por "vpc-" y los nombres de los puntos de conexión de VPC de origen deben empezar por "vpce-".

Tipo: lista

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

Ejemplos

Ejemplo de política de recursos

El siguiente ejemplo bloquea dos direcciones IP y una VPC de origen y permite una AWS cuenta.

YAML

```
Auth:
  ResourcePolicy:
    CustomStatements: [{
      "Effect": "Allow",
      "Principal": "*",
      "Action": "execute-api:Invoke",
      "Resource": "execute-api:/Prod/GET/pets",
      "Condition": {
        "IpAddress": {
          "aws:SourceIp": "1.2.3.4"
        }
      }
    }]

  IpRangeBlacklist:
    - "10.20.30.40"
    - "1.2.3.4"
  SourceVpcBlacklist:
    - "vpce-1a2b3c4d"
  AwsAccountWhitelist:
    - "111122223333"
  IntrinsicVpcBlacklist:
    - "{{resolve:ssm:SomeVPCReference:1}}"
    - !Ref MyVPC
  IntrinsicVpceWhitelist:
    - "{{resolve:ssm:SomeVPCEReference:1}}"
    - !Ref MyVPCE
```

ApiDefinition

Un documento de OpenAPI que define la API.

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
Bucket: String  
Key: String  
Version: String
```

Propiedades

Bucket

Nombre del bucket de Amazon S3 donde se almacena el archivo OpenAPI.

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Bucket](#) propiedad del tipo de `AWS::ApiGateway::RestApi S3Location` datos.

Key

La clave de Amazon S3 del archivo de OpenAPI.

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Key](#) propiedad del tipo de `AWS::ApiGateway::RestApi S3Location` datos.

Version

Para objetos con control de versiones, la versión del archivo de OpenAPI.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Version](#) propiedad del tipo de `AWS::ApiGateway::RestApi S3Location` datos.

Ejemplos

Ejemplo de definición de Uri

Ejemplo de definición de API

YAML

```
DefinitionUri:
  Bucket: amzn-s3-demo-bucket-name
  Key: mykey-name
  Version: 121212
```

CorsConfiguration

Gestione el intercambio de recursos entre orígenes (CORS) para su API Gateway. APIs Especifica el dominio que quieres permitir como cadena o especifica un diccionario con una configuración de Cors adicional.

Note

CORS requiere modificar AWS SAM la definición de OpenAPI. Cree una definición de OpenAPI en línea en `DefinitionBody` para activar CORS. Si `CorsConfiguration` está establecido en la definición de OpenAPI y también en el nivel de propiedad, los AWS SAM fusiona. El nivel de propiedad tiene prioridad sobre la definición de OpenAPI.

Para obtener más información sobre CORS, consulta [Habilitar CORS para un recurso de la API de REST de API Gateway](#) en la Guía para desarrolladores de API Gateway.

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
AllowCredentials: Boolean
AllowHeaders: String
AllowMethods: String
```

```
AllowOrigin: String  
MaxAge: String
```

Propiedades

AllowCredentials

Un valor booleano que indica si se permite que la solicitud contenga credenciales.

Tipo: Booleano

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

AllowHeaders

Cadena de encabezados que se van a permitir.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

AllowMethods

Cadena que contiene los métodos HTTP que se van a permitir.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

AllowOrigin

Cadena de origen que se va a permitir. Puede ser una lista separada por comas en formato de cadena.

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

MaxAge

Cadena que contiene el número de segundos necesarios para almacenar en caché la solicitud CORS Preflight.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

Ejemplos

CorsConfiguration

Ejemplo de configuración de CORS. Esta es solo una parte de un archivo de AWS SAM plantilla que muestra una [AWS::Serverless::API](#) definición con CORS configurado y un [AWS::Serverless::Función](#). Si utiliza una integración de proxy de Lambda o una integración de proxy HTTP, el backend debe devolver los encabezados Access-Control-Allow-Origin, Access-Control-Allow-Methods y Access-Control-Allow-Headers.

YAML

```
Resources:
  ApiGatewayApi:
    Type: AWS::Serverless::Api
    Properties:
      StageName: Prod
      Cors:
        AllowMethods: "'POST, GET'"
        AllowHeaders: "'X-Forwarded-For'"
        AllowOrigin: "'https://example.com'"
        MaxAge: "'600'"
        AllowCredentials: true
      ApiFunction: # Adds a GET method at the root resource via an Api event
      Type: AWS::Serverless::Function
```

```

Properties:
  Events:
    ApiEvent:
      Type: Api
      Properties:
        Path: /
        Method: get
        RestApiId:
          Ref: ApiGatewayApi
Runtime: python3.10
Handler: index.handler
InlineCode: |
  import json
  def handler(event, context):
    return {
      'statusCode': 200,
      'headers': {
        'Access-Control-Allow-Headers': 'Content-Type',
        'Access-Control-Allow-Origin': 'https://example.com',
        'Access-Control-Allow-Methods': 'POST, GET'
      },
      'body': json.dumps('Hello from Lambda!')
    }

```

DomainConfiguration

Configura un dominio personalizado para una API.

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```

AccessAssociation: DomainAccessAssociation
BasePath: List
CertificateArn: String
DomainName: String
EndpointAccessMode: String
EndpointConfiguration: String
MutualTlsAuthentication: MutualTlsAuthentication
NormalizeBasePath: Boolean

```

```
OwnershipVerificationCertificateArn: String  
Policy: Json  
Route53: Route53Configuration  
SecurityPolicy: String
```

Propiedades

AccessAssociation

La configuración necesaria para generar el recurso de `AWS::ApiGateway::DomainNameAccessAssociation`.

AWS SAM genera un [AWS::ApiGateway::DomainNameAccessAssociation](#) recurso cuando se establece esta propiedad. Para obtener información sobre CloudFormation los recursos generados, consulte [Generado CloudFormation recursos para AWS SAM](#).

Tipo: [DomainAccessAssociation](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

BasePath

Una lista de las rutas base que se deben configurar con el nombre de dominio de Amazon API Gateway.

Tipo: lista

Obligatorio: no

Valor predeterminado: /

CloudFormation compatibilidad: esta propiedad es similar a la [BasePath](#) propiedad de un `AWS::ApiGateway::BasePathMapping` recurso. AWS SAM crea varios `AWS::ApiGateway::BasePathMapping` recursos, uno por cada uno de los BasePath especificados en esta propiedad.

CertificateArn

El nombre de recurso de Amazon (ARN) de un certificado AWS administrado del punto final de este nombre de dominio. AWS Gestor de certificados es la única fuente compatible.

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad es similar a la [CertificateArn](#) propiedad de un `AWS::ApiGateway::DomainName` recurso. Si `EndpointConfiguration` se establece en `REGIONAL` (el valor predeterminado), se `CertificateArn` asigna a [RegionalCertificateArn](#) en `AWS::ApiGateway::DomainName`. Si `EndpointConfiguration` está establecido en `EDGE`, se `CertificateArn` asigna a [CertificateArn](#) dentro de `AWS::ApiGateway::DomainName`. Si `EndpointConfiguration` se establece en `PRIVATE`, esta propiedad se pasa al recurso [AWS::ApiGateway::DomainNameV2](#).

Notas adicionales: Para un `EDGE` punto final, debe crear el certificado en la `us-east-1` AWS región.

DomainName

El nombre de dominio personalizado para su API de API Gateway. No se admiten letras mayúsculas.

AWS SAM genera un [AWS::ApiGateway::DomainName](#) recurso cuando se establece esta propiedad. Para obtener información acerca de esta situación, consulta [DomainName se especifica la propiedad](#). Para obtener información sobre CloudFormation los recursos generados, consulte [Generado CloudFormation recursos para AWS SAM](#).

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [DomainName](#) propiedad de un `AWS::ApiGateway::DomainName` recurso o [AWS::ApiGateway::DomainNameV2](#) cuando `EndpointConfiguration` se establece en `PRIVATE`.

EndpointConfiguration

Define el tipo de punto de conexión de API Gateway que se va a asignar al dominio personalizado. El valor de esta propiedad determina cómo se mapea la `CertificateArn` propiedad. CloudFormation

Valores válidos: `EDGE`, `REGIONAL` o `PRIVATE`

Tipo: cadena

Obligatorio: no

Valor predeterminado: REGIONAL

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

EndpointAccessMode

El modo de acceso para el punto final del nombre de dominio personalizado. Se requiere cuando se utilizan políticas de seguridad mejoradas (las que tienen el prefijo `SecurityPolicy_`).

Valores válidos: STRICT o BASIC

Tipo: cadena

Obligatorio: condicional

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [EndpointAccessMode](#) propiedad de un `AWS::ApiGateway::DomainName` recurso o [AWS::ApiGateway::DomainNameV2](#) cuando `EndpointConfiguration` se establece en `PRIVATE`

MutualTlsAuthentication

La configuración de autenticación mutua de seguridad de la capa de transporte (TLS) para un nombre de dominio personalizado.

Tipo: [MutualTlsAuthentication](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [MutualTlsAuthentication](#) propiedad de un `AWS::ApiGateway::DomainName` recurso.

NormalizeBasePath

Indica si se permiten caracteres no alfanuméricos en las rutas base definidas por la propiedad de `BasePath`. Si se establece en `True`, los caracteres no alfanuméricos se eliminan de las rutas base.

Usa `NormalizeBasePath` con la propiedad de `BasePath`.

Tipo: Booleano

Obligatorio: no

Valor predeterminado: True

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

OwnershipVerificationCertificateArn

El ARN del certificado público emitido por ACM para validar la propiedad de su dominio personalizado. Solo se requiere cuando se configura TLS mutua y se especifica un ARN de certificado de CA privado o importado de ACM para CertificateArn.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [OwnershipVerificationCertificateArn](#) propiedad de un `AWS::ApiGateway::DomainName` recurso.

Policy

La política de IAM que se va a asociar al nombre de dominio de API Gateway. Solo es aplicable cuando EndpointConfiguration se establece en PRIVATE.

Tipo: Json

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la Policy propiedad de un `AWS::ApiGateway::DomainNameV2` recurso cuando EndpointConfiguration se establece en PRIVATE. Para ver ejemplos de documentos de política válidos, consulte la [AWS::ApiGateway::DomainNameV2](#).

Route53

Define una configuración de Amazon Route 53.

Tipo: [Route53Configuration](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

SecurityPolicy

La versión de TLS más conjunto de cifrado para este nombre de dominio.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [SecurityPolicy](#) propiedad de un `AWS::ApiGateway::DomainName` recurso o [AWS::ApiGateway::DomainNameV2](#) cuando `EndpointConfiguration` se establece en `PRIVATE`. En los puntos de conexión de `PRIVATE`, solo `TLS_1_2` es compatible.

Ejemplos

DomainName

DomainName ejemplo

YAML

```
Domain:
  DomainName: www.example.com
  CertificateArn: arn-example
  EndpointConfiguration: EDGE
  Route53:
    HostedZoneId: Z1PA6795UKMFR9
  BasePath:
    - foo
    - bar
```

DomainAccessAssociation

Configura una asociación de acceso al dominio entre un origen de asociación de acceso y un nombre de dominio personalizado privado. El único origen de asociación de acceso admitido es un ID de punto de conexión de VPC. Para obtener más información, consulte [Nombres de dominio personalizados para uso privado APIs en API Gateway](#).

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
VpcEndpointId: String
```

Propiedades

VpcEndpointId

El ID de punto de conexión del punto de conexión de interfaz de VPC asociado al servicio de VPC de API Gateway.

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [AccessAssociationSource](#) propiedad de un `AWS::ApiGateway::DomainNameAccessAssociation` recurso.

Route53Configuration

Configura los conjuntos de registros de Route53 para una API.

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
DistributionDomainName: String  
EvaluateTargetHealth: Boolean  
HostedZoneId: String  
HostedZoneName: String  
IPv6: Boolean  
Region: String  
SetIdentifier: String
```

```
VpcEndpointDomainName: String  
VpcEndpointHostedZoneId: String
```

Propiedades

DistributionDomainName

Configura una distribución personalizada del nombre de dominio personalizado de la API.

Tipo: cadena

Obligatorio: no

Predeterminado: utilice la distribución de API Gateway.

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [DNSName](#) propiedad de un `AWS::Route53::RecordSetGroup` `AliasTarget` recurso.

Notas adicionales: El nombre de dominio de una [CloudFront distribución](#).

EvaluateTargetHealth

Si `EvaluateTargetHealth` es verdadero, un registro de alias hereda el estado del AWS recurso al que se hace referencia, como un balanceador de cargas de Elastic Load Balancing u otro registro de la zona alojada.

Tipo: Booleano

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [EvaluateTargetHealth](#) propiedad de un `AWS::Route53::RecordSetGroup` `AliasTarget` recurso.

Notas adicionales: No se puede `EvaluateTargetHealth` establecer en `true` cuando el alias objetivo es una CloudFront distribución.

HostedZoneId

ID de la zona alojada en la que desea crear registros.

Especifica `HostedZoneName` o `HostedZoneId`, pero no ambos. Si tiene múltiples zonas hospedadas con el mismo nombre de dominio, debe especificar la zona hospedada mediante `HostedZoneId`.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [HostedZoneId](#) propiedad de un `AWS::Route53::RecordSetGroup` `RecordSet` recurso.

HostedZoneName

El nombre de la zona alojada en la que desea crear registros.

Especifica `HostedZoneName` o `HostedZoneId`, pero no ambos. Si tiene múltiples zonas hospedadas con el mismo nombre de dominio, debe especificar la zona hospedada mediante `HostedZoneId`.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [HostedZoneName](#) propiedad de un `AWS::Route53::RecordSetGroup` `RecordSet` recurso.

IPv6

Cuando se establece esta propiedad, AWS SAM crea un `AWS::Route53::RecordSet` recurso y establece [Type](#) en AAAA el valor proporcionado `HostedZone`.

Tipo: Booleano

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

Region

Solo conjuntos de registros de recursos basados en la latencia: la EC2 región de Amazon en la que creó el recurso al que hace referencia este conjunto de registros de recursos. El recurso suele ser un AWS recurso, como una EC2 instancia o un balanceador de cargas ELB, y se hace referencia a él mediante una dirección IP o un nombre de dominio DNS, según el tipo de registro.

Cuando Amazon Route 53 recibe una consulta de DNS para un nombre y un tipo de dominio para los que ha creado conjuntos de registros de recursos de latencia, Route 53 selecciona el conjunto

de registros de recursos de latencia que tiene la latencia más baja entre el usuario final y la EC2 región de Amazon asociada. Route 53 devuelve a continuación el valor asociado con el conjunto de registros de recursos seleccionados.

Tenga en cuenta lo siguiente:

- Solo puede especificar un `ResourceRecord` por conjunto de registros de recursos de latencia.
- Solo puedes crear un conjunto de registros de recursos de latencia para cada EC2 región de Amazon.
- No es necesario que crees conjuntos de registros de recursos de latencia para todas las EC2 regiones de Amazon. Route 53 elige la región con la mejor latencia entre las regiones para las que crea conjuntos de registros de recursos de latencia.
- No puede crear conjuntos de registros de recursos sin latencia que tengan los mismos valores `Name` y `Type` que los conjuntos de registros de recursos de latencia.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [Region](#) propiedad de un tipo de `AWS::Route53::RecordSetGroup` `RecordSet` datos.

SetIdentifier

Conjuntos de registros de recursos que tienen una política de direccionamiento que no es simple: Un identificador que diferencia entre varios conjuntos de registros de recursos con la misma combinación de nombre y tipo, como varios conjuntos de registros de recursos ponderados denominados `acme.example.com` de tipo A. En un grupo de conjuntos de registros de recursos que tengan el mismo nombre y tipo, el valor de `SetIdentifier` debe ser exclusivo para cada conjunto de registros de recursos.

Para obtener información sobre cómo dirigir las políticas, consulta [Elección de una política de direccionamiento](#) en la Guía para desarrolladores de Amazon Route 53.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [SetIdentifier](#) propiedad de un tipo de `AWS::Route53::RecordSetGroup` `RecordSet` datos.

VpcEndpointDomainName

Un nombre de DNS del punto de enlace de la interfaz de VPC asociado al servicio de VPC de API Gateway. Esta propiedad solo es obligatoria para los dominios privados.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [DNSName](#) propiedad de un `AWS::Route53::RecordSet AliasTarget` campo.

VpcEndpointHostedZoneId

El ID de zona alojada del punto de conexión de la interfaz de VPC asociado al servicio de VPC de API Gateway. Esta propiedad solo es obligatoria para los dominios privados.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [HostedZoneId](#) propiedad de un `AWS::Route53::RecordSet AliasTarget` campo.

Ejemplos

Ejemplo básico

En este ejemplo, configuramos un dominio personalizado y conjuntos de registros de Route 53 para nuestra API.

YAML

```
Resources:
  MyApi:
    Type: AWS::Serverless::Api
    Properties:
      StageName: Prod
      Domain:
        DomainName: www.example.com
        CertificateArn: arn:aws:acm:us-east-1:123456789012:certificate/
abcdef12-3456-7890-abcd-ef1234567890
        EndpointConfiguration: REGIONAL
```

```
Route53:  
  HostedZoneId: ABCDEFGHIJKLMNOP
```

EndpointConfiguration

El tipo de punto de conexión de una API de REST.

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
IpAddressType: String  
Type: String  
VPCEndpointIds: List
```

Propiedades

IpAddressType

Los tipos de direcciones IP que pueden invocar una API (RestApi).

Valores válidos: `ipv4` o `dualstack`

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [IpAddressType](#) propiedad del tipo de `AWS::ApiGateway::RestApi` EndpointConfiguration datos.

Type

El tipo de punto de conexión de una API de REST.

Valores válidos: `EDGE`, `REGIONAL` o `PRIVATE`

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Types](#) propiedad del tipo de `AWS::ApiGateway::RestApi EndpointConfiguration` datos.

VPCEndpointIds

Una lista de los puntos finales de VPC IDs de una API REST con los que crear los alias de Route53.

Tipo: lista

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [VpcEndpointIds](#) propiedad del tipo de datos. `AWS::ApiGateway::RestApi EndpointConfiguration`

Ejemplos

EndpointConfiguration

Ejemplo de configuraciones de punto de conexión.

YAML

```
EndpointConfiguration:
  Type: PRIVATE
  VPCEndpointIds:
    - vpce-123a123a
    - vpce-321a321a
```

AWS::Serverless::Application

Integrar una aplicación sin servidor desde [AWS Serverless Application Repository](#) o desde un bucket de Amazon S3 como una aplicación anidada. Las aplicaciones anidadas se implementan como recursos anidados [AWS::CloudFormation::Stack](#), que pueden contener varios otros recursos, incluidos otros recursos [AWS::Serverless::Application](#).

Note

Cuando despliega en AWS CloudFormation, AWS SAM transforma sus AWS SAM recursos en CloudFormation recursos. Para obtener más información, consulte [Generado CloudFormation recursos para AWS SAM](#).

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
Type: AWS::Serverless::Application
Properties:
  Location: String | ApplicationLocationObject
  NotificationARNs: List
  Parameters: Map
  Tags: Map
  TimeoutInMinutes: Integer
```

Propiedades

Location

URL de plantilla, ruta de archivo o objeto de ubicación de una aplicación anidada.

Si se proporciona una URL de plantilla, debe seguir el formato especificado en la [CloudFormation TemplateUrl documentación](#) y contener una plantilla SAM CloudFormation o válida.

[ApplicationLocationObject](#) se puede utilizar para especificar una aplicación que se ha publicado en [AWS Serverless Application Repository](#)

Si se proporciona una ruta de archivo local, la plantilla debe seguir el flujo de trabajo que incluye el `sam package` o el comando `sam deploy` para que la aplicación se transforme correctamente.

Tipo: Cadena | [ApplicationLocationObject](#)

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad es similar a la [TemplateURL](#) propiedad de un `AWS::CloudFormation::Stack` recurso. La CloudFormation versión no tarda en [ApplicationLocationObject](#) recuperar una aplicación del AWS Serverless Application Repository.

NotificationARNs

Una lista de temas de Amazon SNS, a la que se envían notificaciones sobre eventos de la pila.

Tipo: lista

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [NotificationARNs](#) propiedad de un `AWS::CloudFormation::Stack` recurso.

Parameters

Valores de los parámetros de aplicación.

Tipo: mapa

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Parameters](#) propiedad de un `AWS::CloudFormation::Stack` recurso.

Tags

Un mapa (cadena a cadena) que especifica las etiquetas agregadas a esta aplicación. Los valores y claves están limitados a caracteres alfanuméricos. Las claves pueden tener de 1 a 127 caracteres Unicode y no pueden ir precedidas de `aws:`. Los valores pueden tener de 1 a 255 caracteres Unicode.

Tipo: mapa

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es similar a la [Tags](#) propiedad de un `AWS::CloudFormation::Stack` recurso. La propiedad `Tags` de SAM consta de pares clave-valor; en CloudFormation ella, consiste en una lista de objetos `Tag`. Cuando se crea la pila, SAM añadirá automáticamente una etiqueta `lambda:createdBy: SAM` a esta aplicación. Además, si esta aplicación es de AWS Serverless Application Repository, SAM también almacenará automáticamente las dos etiquetas `serverlessrepo:applicationId: ApplicationId` adicionales y `serverlessrepo:semanticVersion: SemanticVersion`.

TimeoutInMinutes

El tiempo, en minutos, que se CloudFormation espera a que la pila anidada alcance el `CREATE_COMPLETE` estado. El valor predeterminado es sin tiempo de espera. Cuando CloudFormation detecta que la pila anidada ha alcanzado el `CREATE_COMPLETE` estado,

marca el recurso de la pila anidada como si estuviera CREATE_COMPLETE en la pila principal y reanuda la creación de la pila principal. Si el tiempo de espera expira antes de que llegue la pila anidada CREATE_COMPLETE, CloudFormation marca la pila anidada como fallida y revierte tanto la pila anidada como la pila principal.

Tipo: entero

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [TimeoutInMinutes](#) propiedad de un recurso. AWS::CloudFormation::Stack

valores devueltos

Ref.

Cuando se proporciona el ID lógico de este recurso a la función intrínseca Ref, esta devuelve el nombre del recurso subyacente AWS::CloudFormation::Stack.

Para obtener más información sobre el uso de la función Ref, consulta [Ref](#) en la Guía del usuario de AWS CloudFormation .

Ventilador: GetAtt

Fn::GetAtt devuelve un valor para un atributo especificado de este tipo. A continuación se indican los atributos disponibles y los valores devueltos de muestra.

Para obtener más información acerca de cómo usar Fn::GetAtt, consulta [Fn::GetAtt](#) en la Guía del usuario de AWS CloudFormation .

Outputs.ApplicationOutputName

El valor de la salida de la pila con su nombre *ApplicationOutputName*.

Ejemplos

Aplicaciones SAR

Aplicación que usa una plantilla del Serverless Application Repository

YAML

```
Type: AWS::Serverless::Application
Properties:
  Location:
    ApplicationId: 'arn:aws:serverlessrepo:us-east-1:012345678901:applications/my-
application'
    SemanticVersion: 1.0.0
  Parameters:
    StringParameter: parameter-value
    IntegerParameter: 2
```

Aplicación normal

Aplicación desde una URL de S3

YAML

```
Type: AWS::Serverless::Application
Properties:
  Location: https://s3.amazonaws.com/sam-s3-demo-bucket/template.yaml
```

ApplicationLocationObject

Una solicitud que se ha publicado en [AWS Serverless Application Repository](#).

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
ApplicationId: String
SemanticVersion: String
```

Propiedades

ApplicationId

El nombre de recurso de Amazon (ARN) de la aplicación.

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

SemanticVersion

La versión semántica de la aplicación.

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

Ejemplos

my-application

Ejemplo de aplicación de ubicación de un objeto

YAML

```
Location:
  ApplicationId: 'arn:aws:serverlessrepo:us-east-1:012345678901:applications/my-
application'
  SemanticVersion: 1.0.0
```

AWS::Serverless::CapacityProvider

Crea un proveedor de capacidad para AWS Lambda Event Functions que permite ejecutar instancias gestionadas por Lambda en instancias de Amazon Elastic Compute Cloud propiedad del cliente. Este recurso forma parte de la función de instancias gestionadas por Lambda, que optimiza los costes de las cargas de trabajo de Lambda a gran escala mediante el uso de modelos de precios de Amazon EC2.

El proveedor de capacidad administra el ciclo de vida de las instancias de Amazon EC2 y proporciona la infraestructura necesaria para que las funciones de Lambda se ejecuten en los

recursos informáticos propiedad del cliente, al tiempo que mantiene el modelo de programación sin servidor.

Note

Cuando las implementa AWS CloudFormation, AWS SAM transforma sus recursos en recursos. AWS SAM CloudFormation Para obtener más información, consulte [Generado CloudFormation recursos para AWS SAM](#).

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
Type: AWS::Serverless::CapacityProvider
Properties:
  CapacityProviderName: String
  VpcConfig: VpcConfig
  OperatorRole: String
  Tags: Map
  PropagateTags: Boolean
  InstanceRequirements: InstanceRequirements
  ScalingConfig: ScalingConfig
  LoggingConfig: CapacityProviderLoggingConfig
  KmsKeyArn: String
  ManagedResourceTags: ManagedResourceTags
```

Propiedades

CapacityProviderName

El nombre del proveedor de capacidad. Este nombre debe ser único en tu AWS cuenta y región.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [CapacityProviderName](#) propiedad de un `AWS::Lambda::CapacityProvider` recurso.

VpcConfig

La configuración de VPC para el proveedor de capacidad. Especifica las subredes de VPC y los grupos de seguridad en los que se lanzarán las instancias de Amazon EC2.

Tipo: [VpcConfig](#)

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [VpcConfig](#) propiedad de un recurso. `AWS::Lambda::CapacityProvider`

OperatorRole

El ARN del rol de operador de Lambda con permisos para crear y administrar instancias de Amazon EC2 y los recursos relacionados en la cuenta del cliente. Si no se proporciona, genera AWS SAM automáticamente un rol con los permisos necesarios.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [CapacityProviderOperatorRoleArn](#) propiedad [PermissionsConfig](#) de un `AWS::Lambda::CapacityProvider` recurso.

Tags

Un mapa de pares clave-valor para aplicarlo al proveedor de capacidad y sus recursos asociados.

Tipo: mapa

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [Tags](#) propiedad de un `AWS::Lambda::CapacityProvider` recurso. La Tags propiedad se AWS SAM compone de pares clave-valor (mientras que en CloudFormation esta propiedad se compone de una lista de objetos Tag). Además, agrega AWS SAM automáticamente una `lambda:createdBy:SAM` etiqueta a esta función Lambda y a las funciones predeterminadas generadas para esta función.

PropagateTags

Indica si se deben pasar o no las etiquetas de la propiedad Tags a los recursos `AWS::Serverless::CapacityProvider` generados. TrueConfigúrelo para propagar las etiquetas en los recursos generados.

Tipo: Booleano

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

InstanceRequirements

Especificaciones de los tipos de instancias informáticas que puede usar el proveedor de capacidad. Esto incluye los requisitos de arquitectura `allowed` o los tipos de `excluded` instancias.

Tipo: [InstanceRequirements](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [InstanceRequirements](#) propiedad de un `AWS::Lambda::CapacityProvider` recurso.

ScalingConfig

La configuración de escalado del proveedor de capacidad. Define cómo el proveedor de capacidad escala las instancias de Amazon EC2 en función de la demanda.

Tipo: [ScalingConfig](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [CapacityProviderScalingConfig](#) propiedad de un `AWS::Lambda::CapacityProvider` recurso.

LoggingConfig

Se utiliza `LoggingConfig` para configurar el registro del sistema para el proveedor de capacidad. Los registros del sistema capturan la actividad de escalado y los eventos operativos y los envían a un grupo de CloudWatch registros de Amazon a un nivel de registro configurable.

Tipo: [CapacityProviderLoggingConfig](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [LoggingConfig](#) propiedad [TelemetryConfig](#) de un `AWS::Lambda::CapacityProvider` recurso.

KmsKeyArn

El ARN de la AWS KMS clave utilizada para cifrar los datos en reposo y en tránsito para el proveedor de capacidad.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [KmsKeyArn](#) propiedad de un `AWS::Lambda::CapacityProvider` recurso.

ManagedResourceTags

Aplica etiquetas explícitas a las instancias de Amazon Elastic Compute Cloud y a otros recursos gestionados que el proveedor de capacidad aprovisiona en tu nombre.

Tipo: [ManagedResourceTags](#)

Obligatorio: no

CloudFormation compatibilidad: AWS SAM usa esta propiedad para construir la [PropagateTags](#) propiedad de un `AWS::Lambda::CapacityProvider` recurso.

valores devueltos

Ref.

Cuando se proporciona el identificador lógico de este recurso a la función Ref intrínseca, devuelve el nombre del proveedor de capacidad.

Para obtener más información sobre el uso de la función Ref, consulta [Ref](#) en la Guía del usuario de AWS CloudFormation .

Fn::GetAtt

Fn::GetAtt devuelve un valor para un atributo especificado de este tipo. A continuación se indican los atributos disponibles y los valores devueltos de muestra.

Para obtener más información acerca de cómo usar Fn::GetAtt, consulta [Fn::GetAtt](#) en la Guía del usuario de AWS CloudFormation .

Arn

El ARN del proveedor de capacidad.

Ejemplos

Proveedor de capacidad básica

En el siguiente ejemplo, se crea un proveedor de capacidad básico con una configuración de VPC.

```
MyCapacityProvider:
  Type: AWS::Serverless::CapacityProvider
  Properties:
    CapacityProviderName: my-capacity-provider
    VpcConfig:
      SubnetIds:
        - subnet-12345678
        - subnet-87654321
      SecurityGroupIds:
        - sg-12345678
    Tags:
      Environment: Production
      Team: ServerlessTeam
```

Proveedor de capacidad avanzado con escalado

En el siguiente ejemplo, se crea un proveedor de capacidad con requisitos de instancia personalizados y una configuración de escalado.

```
AdvancedCapacityProvider:
  Type: AWS::Serverless::CapacityProvider
  Properties:
    CapacityProviderName: advanced-capacity-provider
    VpcConfig:
      SubnetIds:
        - subnet-12345678
        - subnet-87654321
      SecurityGroupIds:
        - sg-12345678
    OperatorRole: arn:aws:iam::123456789012:role/MyCapacityProviderRole
    PropagateTags: true
    InstanceRequirements:
```

```

Architectures:
  - x86_64
ExcludedTypes:
  - t2.micro
ScalingConfig:
  MaxVCpuCount: 10
  AverageCPUUtilization: 70.0
KmsKeyArn: arn:aws:kms:us-
east-1:123456789012:key/12345678-1234-1234-1234-123456789012
LoggingConfig:
  LogGroup: /aws/lambda/capacity-provider/advanced-capacity-provider
  SystemLogLevel: INFO
ManagedResourceTags:
  Tags:
    CostCenter: cc-1234
    Environment: Production
Tags:
  Environment: Production
  CostCenter: Engineering

```

VpcConfig

Configura la configuración de la VPC para un proveedor de capacidad, incluidas las subredes y los grupos de seguridad EC2 donde se lanzarán las instancias.

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```

SubnetIds: List
SecurityGroupIds: List

```

Propiedades

SubnetIds

Una lista de subredes en las que IDs se lanzarán EC2 las instancias. Debe especificarse al menos una subred.

Tipo: lista

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [SubnetIds](#) propiedad [VpcConfig](#) de un `AWS::Lambda::CapacityProvider` recurso.

SecurityGroupIds

Una lista de grupos de seguridad IDs para asociarlos a las EC2 instancias. Si no se especifica, se utilizará el grupo de seguridad predeterminado para la VPC.

Tipo: lista

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [SecurityGroupIds](#) [VpcConfig](#) propiedad de un `AWS::Lambda::CapacityProvider` recurso.

Ejemplos

Configuración de la VPC

El siguiente ejemplo muestra una configuración de VPC con varias subredes y grupos de seguridad.

```
VpcConfig:
  SubnetIds:
    - subnet-12345678
    - subnet-87654321
  SecurityGroupIds:
    - sg-12345678
    - sg-87654321
```

InstanceRequirements

Especifica los requisitos de EC2 las instancias que lanzará el proveedor de capacidad, incluidas las arquitecturas y las restricciones de tipo de instancia.

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
Architectures: List  
AllowedTypes: List  
ExcludedTypes: List
```

Note

Puede especificar una de las dos `AllowedTypes` opciones `ExcludedTypes` al definir los requisitos de instancia para su proveedor de capacidad, pero no ambas opciones.

Propiedades

Architectures

Las instrucciones configuran las arquitecturas para las instancias del proveedor de capacidad.

Valores válidos: `x86_64` o `arm64`

Tipo: lista

Obligatorio: no

Valor predeterminado: `x86_64`

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [Architectures](#) propiedad [InstanceRequirements](#) de un `AWS::Lambda::CapacityProvider` recurso.

AllowedTypes

Una lista de los tipos de EC2 instancias permitidos para la instancia del proveedor de capacidad.

Tipo: lista

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [AllowedInstanceTypes](#) propiedad [InstanceRequirements](#) de un `AWS::Lambda::CapacityProvider` recurso.

ExcludedTypes

Una lista de tipos de EC2 instancias para excluirlos del proveedor de capacidad.

Tipo: lista

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [ExcludedInstanceTypes](#) propiedad [InstanceRequirements](#) de un `AWS::Lambda::CapacityProvider` recurso.

Ejemplos

Configuración de requisitos de instancia

El siguiente ejemplo muestra los requisitos de la instancia con restricciones específicas de arquitectura y tipo de instancia.

```
InstanceRequirements:
  Architectures:
    - x86_64
  ExcludedTypes:
    - t2.micro
```

ScalingConfig

Especifica cómo el proveedor de capacidad escala las instancias de Amazon Elastic Compute Cloud (Amazon EC2) en función de la demanda. Debe establecer la cantidad máxima de vCPU que el proveedor de capacidad puede aprovisionar y el objetivo de uso medio de la CPU que desencadena el escalado.

Sintaxis

Para declarar esta entidad en la plantilla AWS Serverless Application Model (AWS SAM), utilice la siguiente sintaxis.

YAML

```
MaxVCpuCount: Integer
AverageCPUUtilization: Double
```

Propiedades

MaxVCpuCount

La cantidad máxima de vCPU que el proveedor de capacidad puede aprovisionar en todas las instancias de procesamiento.

Tipo: entero

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [MaxVCpuCount](#) propiedad [CapacityProviderScalingConfig](#) de un `AWS::Lambda::CapacityProvider` recurso.

AverageCPUUtilization

El porcentaje promedio de uso de la CPU objetivo (0-100) para las decisiones de escalado. Cuando la utilización media de la CPU supere este umbral, el proveedor de capacidad ampliará las instancias de Amazon EC2. Cuando se especifica, AWS SAM crea un [CapacityProviderScalingConfig](#) `AWS::Lambda::CapacityProvider` recurso con los valores establecidos en 'Manual' y [ScalingMode](#) [ScalingPolicies](#) establecidos en. `[{PredefinedMetricType: 'LambdaCapacityProviderAverageCPUUtilization', TargetValue: <this value>}]`

Tipo: Doble

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

Ejemplos

Scaling configuration (Escalado de configuración)

El siguiente ejemplo muestra una configuración de escalado con un número máximo de vCPU y un uso medio de la CPU.

```
ScalingConfig:
  MaxVCpuCount: 10
```



```
AverageCPUUtilization: 70.0
```

ManagedResourceTags

Aplica etiquetas explícitas a las instancias de Amazon Elastic Compute Cloud y a otros recursos que el proveedor de capacidad aprovisiona y administra en su nombre.

Note

La `ManagedResourceTags` propiedad controla el etiquetado de las instancias de Amazon EC2 y otros recursos que Lambda administra para el proveedor de capacidad. Esta propiedad difiere de la [PropagateTags](#) propiedad, que especifica si la [propiedad](#) `Tags` AWS SAM se propaga a los CloudFormation recursos que se AWS SAM generan a partir de la plantilla.

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
Tags: Map
```

Propiedades

Se admiten las siguientes propiedades `ManagedResourceTags`.

Tags

Un mapa de pares clave-valor para aplicarlo a las instancias de Amazon EC2 y a otros recursos que administra el proveedor de capacidad.

Tipo: mapa

Obligatorio: no

CloudFormation compatibilidad: AWS SAM usa esta propiedad para construir la [PropagateTags](#) propiedad de un recurso. `AWS::Lambda::CapacityProvider` AWS SAM establece `Mode` `Explicit` y convierte los pares clave-valor en una lista de objetos `Tag` en. `ExplicitTags`

Ejemplos

Aplica etiquetas explícitas a los recursos gestionados

El siguiente ejemplo aplica un conjunto explícito de etiquetas a las instancias de Amazon EC2 que administra el proveedor de capacidad.

```
ManagedResourceTags:
  Tags:
    CostCenter: cc-1234
    Environment: Production
```

AWS::Serverless::Connector

Configura permisos entre dos recursos. Para obtener una introducción a los conectores, consulta [Administrar los permisos de recursos con AWS SAM conectores](#).

Para obtener más información sobre AWS CloudFormation los recursos generados, consulte [CloudFormation recursos generados cuando se especifica AWS::Serverless::Connector](#).

Para enviar comentarios sobre los conectores, [envíe un nuevo número al repositorio AWS](#) GitHub serverless-application-model.

Note

Cuando lo despliegas AWS CloudFormation, AWS SAM transforma tus recursos en recursos. AWS SAM CloudFormation Para obtener más información, consulte [Generado CloudFormation recursos para AWS SAM](#).

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa cualquiera de las siguientes sintaxis.

Note

Recomendamos utilizar la sintaxis de los conectores integrados en la mayoría de los casos de uso. Al estar integrado en el recurso de origen, es más fácil de leer y mantener a lo largo del tiempo. Cuando necesites hacer referencia a un recurso fuente que no esté dentro de la

misma AWS SAM plantilla, como un recurso de una pila anidada o un recurso compartido, usa la `AWS::Serverless::Connector` sintaxis.

Conectores integrados

`<source-resource-logical-id>:`

Connectors:

`<connector-logical-id>:`

Properties:

Destination: [ResourceReference](#) | *List of* [ResourceReference](#)

Permissions: *List*

SourceReference: [SourceReference](#)

AWS::Serverless:: Conector

Type: `AWS::Serverless::Connector`

Properties:

Destination: [ResourceReference](#) | *List of* [ResourceReference](#)

Permissions: *List*

Source: [ResourceReference](#)

Propiedades

Destination

El recurso de destino.

Tipo: [ResourceReference](#) | Lista de [ResourceReference](#)

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

Permissions

El tipo de permiso que el recurso de origen puede realizar en el recurso de destino.

Read incluye acciones AWS Identity and Access Management (IAM) que permiten leer los datos del recurso.

Write incluye acciones de IAM que permiten iniciar y escribir datos en un recurso.

Valores válidos: `Read` o `Write`

Tipo: lista

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

Source

El recurso fuente. Obligatorio cuando se utiliza la sintaxis `AWS::Serverless::Connector`.

Tipo: [ResourceReference](#)

Obligatorio: condicional

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

SourceReference

El recurso fuente.

Note

Utilícelo con la sintaxis de los conectores integrados al definir propiedades adicionales para el recurso de origen.

Tipo: [SourceReference](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

Ejemplos

Conectores integrados

En el siguiente ejemplo, se utilizan conectores integrados para definir una conexión de `Write` datos entre una función de AWS Lambda eventos y una tabla de Amazon DynamoDB:

```
Transform: AWS::Serverless-2016-10-31
...
Resources:
  MyTable:
    Type: AWS::Serverless::SimpleTable
  MyFunction:
    Type: AWS::Serverless::Function
  Connectors:
    MyConn:
      Properties:
        Destination:
          Id: MyTable
        Permissions:
          - Write
    ...
```

En el siguiente ejemplo, se utilizan conectores integrados para definir los Read y los permisos Write:

```
Transform: AWS::Serverless-2016-10-31
...
Resources:
  MyFunction:
    Type: AWS::Serverless::Function
  Connectors:
    MyConn:
      Properties:
        Destination:
          Id: MyTable
        Permissions:
          - Read
          - Write
  MyTable:
    Type: AWS::DynamoDB::Table
    ...
```

En el siguiente ejemplo, se utilizan conectores integrados para definir un recurso fuente con una propiedad distinta de Id:

```
Transform: AWS::Serverless-2016-10-31
Transform: AWS::Serverless-2016-10-31
...
```

```
Resources:
  MyApi:
    Type: AWS::Serverless::Api
    Connectors:
      ApitoLambdaConn:
        Properties:
          SourceReference:
            Qualifier: Prod/GET/foobar
          Destination:
            Id: MyTable
          Permissions:
            - Read
            - Write
  MyTable:
    Type: AWS::DynamoDB::Table
    ...
```

AWS::Serverless: :Conector

En el siguiente ejemplo, se utiliza el [AWS::Serverless: :Conector](#) recurso para que una AWS Lambda función lea y escriba en una tabla de Amazon DynamoDB:

```
MyConnector:
  Type: AWS::Serverless::Connector
  Properties:
    Source:
      Id: MyFunction
    Destination:
      Id: MyTable
    Permissions:
      - Read
      - Write
```

En el siguiente ejemplo, se utiliza el recurso [AWS::Serverless: :Conector](#) para que una función de Lambda escriba en un tema de Amazon SNS, con ambos recursos en la misma plantilla:

```
MyConnector:
  Type: AWS::Serverless::Connector
  Properties:
    Source:
      Id: MyLambda
    Destination:
      Id: MySNSTopic
```

Permissions:

- Write

En el siguiente ejemplo, se utiliza el recurso [AWS::Serverless::Conector](#) para que un tema de Amazon SNS se escriba en una función de Lambda y, a continuación, en una tabla de Amazon DynamoDB, con todos los recursos en la misma plantilla:

```
Transform: AWS::Serverless-2016-10-31
Resources:
  Topic:
    Type: AWS::SNS::Topic
    Properties:
      Subscription:
        - Endpoint: !GetAtt Function.Arn
          Protocol: lambda

  Function:
    Type: AWS::Serverless::Function
    Properties:
      Runtime: nodejs16.x
      Handler: index.handler
      InlineCode: |
        const AWS = require('aws-sdk');
        exports.handler = async (event, context) => {
          const docClient = new AWS.DynamoDB.DocumentClient();
          await docClient.put({
            TableName: process.env.TABLE_NAME,
            Item: {
              id: context.awsRequestId,
              event: JSON.stringify(event)
            }
          }).promise();
        };
      Environment:
        Variables:
          TABLE_NAME: !Ref Table

  Table:
    Type: AWS::Serverless::SimpleTable

  TopicToFunctionConnector:
    Type: AWS::Serverless::Connector
    Properties:
```

```

Source:
  Id: Topic
Destination:
  Id: Function
Permissions:
  - Write

FunctionToTableConnector:
  Type: AWS::Serverless::Connector
  Properties:
    Source:
      Id: Function
    Destination:
      Id: Table
    Permissions:
      - Write

```

La siguiente es la AWS CloudFormation plantilla transformada del ejemplo anterior:

```

"FunctionToTableConnectorPolicy": {
  "Type": "AWS::IAM::ManagedPolicy",
  "Metadata": {
    "aws:sam:connectors": {
      "FunctionToTableConnector": {
        "Source": {
          "Type": "AWS::Lambda::Function"
        },
        "Destination": {
          "Type": "AWS::DynamoDB::Table"
        }
      }
    }
  },
  "Properties": {
    "PolicyDocument": {
      "Version": "2012-10-17",
      "Statement": [
        {
          "Effect": "Allow",
          "Action": [
            "dynamodb:PutItem",
            "dynamodb:UpdateItem",
            "dynamodb:DeleteItem",

```



```

        "dynamodb:BatchWriteItem",
        "dynamodb: PartiQLDelete",
        "dynamodb: PartiQLInsert",
        "dynamodb: PartiQLUpdate"
    ],
    "Resource": [
        {
            "Fn::GetAtt": [
                "MyTable",
                "Arn"
            ]
        },
        {
            "Fn::Sub": [
                "${DestinationArn}/index/*",
                {
                    "DestinationArn": {
                        "Fn::GetAtt": [
                            "MyTable",
                            "Arn"
                        ]
                    }
                }
            ]
        }
    ]
}
]
}
],
"Roles": [
    {
        "Ref": "MyFunctionRole"
    }
]
}
}
}

```

ResourceReference

Referencia a un recurso que utiliza el tipo de recurso [AWS::Serverless::Conector](#).

 Note

Para los recursos de la misma plantilla, proporciona la Id. Para recursos que no estén dentro de la misma plantilla, utilice una combinación de otras propiedades. Para obtener más información, consulta [AWS SAM referencia de conector](#).

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
Arn: String  
Id: String  
Name: String  
Qualifier: String  
QueueUrl: String  
ResourceId: String  
RoleName: String  
Type: String
```

Propiedades

Arn

El ARN de un recurso .

Tipo: cadena

Obligatorio: condicional

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

Id

El [ID lógico](#) de un recurso de la misma plantilla.

 Note

Si `Id` se especifica, si el conector genera políticas AWS Identity and Access Management (IAM), la función de IAM asociada a esas políticas se deducirá del recurso. `Id` Cuando `Id` no se especifica, proporciona `RoleName` del recurso para que los conectores adjunten las políticas de IAM generadas a un rol de IAM.

Tipo: cadena

Obligatorio: condicional

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un equivalente. CloudFormation

Name

El nombre de un recurso .

Tipo: cadena

Obligatorio: condicional

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

Qualifier

Calificador de un recurso que reduce su alcance. `Qualifier` reemplaza el valor `*` al final de un ARN de restricción de recursos. Para ver un ejemplo, consulta [API Gateway que invoca una función de evento de Lambda](#).

 Note

La definición del calificador varía según el tipo de recurso. Para ver una lista los puntos de enlace de origen y destino admitidos, consulta [AWS SAM referencia de conector](#).

Tipo: cadena

Obligatorio: condicional

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

QueueUrl

La URL de la cola de Amazon SQS. Esta propiedad solo se aplica a los recursos de Amazon SQS.

Tipo: cadena

Obligatorio: condicional

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

ResourceId

El ID de un recurso. Por ejemplo, el ID de API de la API Gateway.

Tipo: cadena

Obligatorio: condicional

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

RoleName

El nombre del rol asociado a un recurso.

Note

Cuando se especifica `Id`, si el conector genera políticas de IAM, el rol de IAM asociado a esas políticas se deducirá del recurso de `Id`. Cuando `Id` no se especifica, proporciona `RoleName` del recurso para que los conectores adjunten las políticas de IAM generadas a un rol de IAM.

Tipo: cadena

Obligatorio: condicional

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

Type

El CloudFormation tipo de recurso. Para obtener más información, consulta la [Referencia de tipos de recursos y propiedades de AWS](#).

Tipo: cadena

Obligatorio: condicional

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

Ejemplos

API Gateway que invoca una función de evento de Lambda

En el siguiente ejemplo, se utiliza el [AWS::Serverless::Conector](#) recurso para permitir que Amazon API Gateway invoque una función de AWS Lambda evento.

YAML

```
Transform: AWS::Serverless-2016-10-31
Resources:
  MyRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Statement:
          - Effect: Allow
            Action: sts:AssumeRole
            Principal:
              Service: lambda.amazonaws.com
      ManagedPolicyArns:
        - !Sub arn:${AWS::Partition}:iam::aws:policy/service-role/
          AWSLambdaBasicExecutionRole

  MyFunction:
    Type: AWS::Lambda::Function
    Properties:
      Role: !GetAtt MyRole.Arn
      Runtime: nodejs16.x
      Handler: index.handler
      Code:
```

```

ZipFile: |
  exports.handler = async (event) => {
    return {
      statusCode: 200,
      body: JSON.stringify({
        "message": "It works!"
      }),
    };
  };
};

```

MyApi:

```

Type: AWS::ApiGatewayV2::Api
Properties:
  Name: MyApi
  ProtocolType: HTTP

```

MyStage:

```

Type: AWS::ApiGatewayV2::Stage
Properties:
  ApiId: !Ref MyApi
  StageName: prod
  AutoDeploy: True

```

MyIntegration:

```

Type: AWS::ApiGatewayV2::Integration
Properties:
  ApiId: !Ref MyApi
  IntegrationType: AWS_PROXY
  IntegrationUri: !Sub arn:aws:apigateway:${AWS::Region}:lambda:path/2015-03-31/
functions/${MyFunction.Arn}/invocations
  IntegrationMethod: POST
  PayloadFormatVersion: "2.0"

```

MyRoute:

```

Type: AWS::ApiGatewayV2::Route
Properties:
  ApiId: !Ref MyApi
  RouteKey: GET /hello
  Target: !Sub integrations/${MyIntegration}

```

MyConnector:

```

Type: AWS::Serverless::Connector
Properties:
  Source: # Use 'Id' when resource is in the same template

```

```
Type: AWS::ApiGatewayV2::Api
ResourceId: !Ref MyApi
Qualifier: prod/GET/hello # Or "*" to allow all routes
Destination: # Use 'Id' when resource is in the same template
Type: AWS::Lambda::Function
Arn: !GetAtt MyFunction.Arn
Permissions:
  - Write

Outputs:
  Endpoint:
    Value: !Sub https://${MyApi}.execute-api.${AWS::Region}.${AWS::URLSuffix}/prod/hello
```

SourceReference

Referencia a un recurso de origen que utiliza el tipo de recurso [AWS::Serverless::Conector](#).

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
Qualifier: String
```

Propiedades

Qualifier

Calificador de un recurso que reduce su alcance. `Qualifier` reemplaza el valor `*` al final de un ARN de restricción de recursos.

Note

La definición del calificador varía según el tipo de recurso. Para ver una lista los puntos de enlace de origen y destino admitidos, consulta [AWS SAM referencia de conector](#).

Tipo: cadena

Obligatorio: condicional

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

Ejemplos

En el siguiente ejemplo, se utilizan conectores integrados para definir un recurso fuente con una propiedad distinta de **Id**:

```
Transform: AWS::Serverless-2016-10-31
...
Resources:
  MyApi:
    Type: AWS::Serverless::Api
    Connectors:
      ApitoLambdaConn:
        Properties:
          SourceReference:
            Qualifier: Prod/GET/foobar
          Destination:
            Id: MyTable
          Permissions:
            - Read
            - Write
  MyTable:
    Type: AWS::DynamoDB::Table
    ...
```

AWS::Serverless::Función

El `AWS::Serverless::Function` recurso crea una función de eventos de Lambda, un rol de ejecución AWS Identity and Access Management (IAM) y asignaciones de fuentes de eventos que activan la función.

Función de eventos es el nombre de la documentación del tipo de función Lambda original, anteriormente conocido como funciones Lambda. El tipo de `AWS::Serverless::Function` recurso, el `AWS::Lambda::Function` recurso que genera y sus propiedades siguen utilizando la función, por lo que no es necesario cambiar las plantillas.

El [AWS::Serverless::Function](#) recurso también admite el atributo Metadata resource, por lo que puede indicarle que cree AWS SAM los tiempos de ejecución personalizados que requiera su aplicación. Para obtener más información acerca de la creación de tiempos de ejecución personalizados, consulta [Creación de funciones de eventos de Lambda con tiempos de ejecución personalizados en AWS SAM](#).

Note

Al realizar la implementación en AWS CloudFormation, AWS SAM transforma AWS SAM los recursos en CloudFormation recursos. Para obtener más información, consulte [Generado CloudFormation recursos para AWS SAM](#).

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
Type: AWS::Serverless::Function
Properties:
  Architectures: List
  AssumeRolePolicyDocument: JSON
  AutoPublishAlias: String
  AutoPublishAliasAllProperties: Boolean
  AutoPublishCodeSha256: String
  CapacityProviderConfig: CapacityProviderConfig
  CodeSigningConfigArn: String
  CodeUri: String | FunctionCode
  DeadLetterQueue: Map | DeadLetterQueue
  DeploymentPreference: DeploymentPreference
  Description: String
  DurableConfig: DurableConfig
  Environment: Environment
  EphemeralStorage: EphemeralStorage
  EventInvokeConfig: EventInvokeConfiguration
  Events: EventSource
  FileSystemConfigs: List
  FunctionName: String
  FunctionScalingConfig: FunctionScalingConfig
```

```

FunctionUrlConfig: FunctionUrlConfig
Handler: String
ImageConfig: ImageConfig
ImageUri: String
InlineCode: String
KmsKeyArn: String
Layers: List
LoggingConfig: LoggingConfig
MemorySize: Integer
PackageType: String
PermissionsBoundary: String
Policies: String | List | Map
PublishToLatestPublished: Boolean
PropagateTags: Boolean
ProvisionedConcurrencyConfig: ProvisionedConcurrencyConfig
RecursiveLoop: String
ReservedConcurrentExecutions: Integer
Role: String
RolePath: String
Runtime: String
RuntimeManagementConfig: RuntimeManagementConfig
SnapStart: SnapStart
SourceKMSKeyArn: String
Tags: Map
TenancyConfig: TenancyConfig
Timeout: Integer
Tracing: String
VersionDescription: String
VersionDeletionPolicy: String
VpcConfig: VpcConfig

```

Propiedades

Architectures

La arquitectura del conjunto de instrucciones para la función.

Para obtener más información sobre esta propiedad, consulta [Arquitecturas del conjunto de instrucciones Lambda](#) en la Guía para desarrolladores de AWS Lambda .

Valores válidos: uno de los valores x86_64 o arm64.

Tipo: lista

Obligatorio: no

Valor predeterminado: x86_64

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Architectures](#) propiedad de un `AWS::Lambda::Function` recurso.

AssumeRolePolicyDocument

Añade un AssumeRolePolicyDocument formulario al valor predeterminado creado Role para esta función. Si no se especifica esta propiedad, AWS SAM agrega una función de asunción predeterminada para esta función.

Tipo: JSON

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es similar a la [AssumeRolePolicyDocument](#) propiedad de un `AWS::IAM::Role` recurso. AWS SAM añade esta propiedad al rol de IAM generado para esta función. Si se proporciona el nombre de recurso de Amazon (ARN) de un rol para esta función, esta propiedad no tiene efecto.

AutoPublishAlias

El nombre del alias de Lambda. Para obtener más información acerca de los alias en Lambda, consulta [Alias de función de Lambda](#) en la Guía para desarrolladores de AWS Lambda . Para ver ejemplos que utilizan esta propiedad, consulta [Implementación gradual de aplicaciones sin servidor con AWS SAM](#).

AWS SAM genera `AWS::Lambda::Version` y proporciona `AWS::Lambda::Alias` recursos cuando se establece esta propiedad. Para obtener información acerca de esta situación, consulta [AutoPublishAlias se especifica la propiedad](#). Para obtener información general sobre CloudFormation los recursos generados, consulte [Generado CloudFormation recursos para AWS SAM](#).

Publicación de versiones y funciones intrínsecas

AWS SAM determina si se debe publicar una nueva versión comparando los valores de las propiedades de la plantilla entre las implementaciones. En el caso de los valores proporcionados a través de funciones intrínsecas, AWS SAM compara la referencia no resuelta, no el valor resuelto. Un ejemplo es una variable de entorno establecida como `!Ref` de un parámetro de plantilla.

Si una implementación solo cambia el valor de un parámetro de pila, la plantilla en sí no se modifica. En este caso, AWS SAM no publica una nueva versión ni actualiza el alias. AWS SAM aplica el cambio únicamente a la `$LATEST` versión de la función. Para obtener soluciones alternativas, consulte el ejemplo de uso. [AWS::LanguageExtensions](#)

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

AutoPublishAliasAllProperties

Especifica cuándo se crea una nueva [AWS::Lambda::Version](#). Cuando es `true`, se crea una nueva versión de Lambda cuando se modifica cualquier propiedad de la función de Lambda. Cuando es `false`, se crea una nueva versión de Lambda solo cuando se modifica alguna de las siguientes propiedades:

- `Environment`, `MemorySize`, o `SnapStart`.
- Cualquier cambio que dé lugar a una actualización de la propiedad `Code`, como `CodeDict`, `ImageUri` o `InlineCode`.

Cambios de parámetros y publicación de versiones

AWS SAM publica una nueva versión solo cuando el valor de una propiedad cambia en la plantilla. Si el valor de una propiedad proviene de una función intrínseca, AWS SAM ve la referencia no resuelta. Un ejemplo es una variable de entorno definida como un parámetro `!Ref` de plantilla. Las implementaciones que solo cambian el valor del parámetro no publican una nueva versión, ni siquiera con `AutoPublishAliasAllProperties: true`. AWS SAM aplica el cambio `$LATEST` únicamente a.

Utilice una de las siguientes soluciones para forzar AWS SAM la publicación de una nueva versión cuando solo cambie el valor de un parámetro:

- Estándar AWS SAM e CloudFormation implementaciones: añada la [AWS::LanguageExtensions transformación](#) anterior `AWS::Serverless-2016-10-31` en la sección de tu plantilla. Transform CloudFormation resuelve las referencias a sus valores reales antes de AWS SAM

calcular el hash de la versión. A continuación, un cambio en el valor del parámetro publica una nueva versión y actualiza el alias:

Transform:

- AWS::LanguageExtensions
- AWS::Serverless-2016-10-31

- AWS Serverless Application Repository aplicaciones: solo AWS Serverless Application Repository admite la `AWS::Serverless-2016-10-31` transformación y rechaza las plantillas que declaran `AWS::LanguageExtensions`. En el caso de las aplicaciones que publique a través de AWS Serverless Application Repository, introduzca el valor directamente como un literal en la plantilla, en lugar de hacer referencia a un parámetro. Esto hace que el cambio sea visible al AWS SAM calcular el hash de la versión.

La propiedad requiere que `AutoPublishAlias` se defina.

Si también se especifica `AutoPublishCodeSha256`, su comportamiento tiene prioridad sobre `AutoPublishAliasAllProperties: true`.

Tipo: Booleano

Obligatorio: no

Valor predeterminado: `false`

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

`AutoPublishCodeSha256`

Cuando se usa, esta cadena funciona con el valor `CodeUri` para determinar si es necesario publicar una nueva versión de Lambda. Esta propiedad se suele utilizar para resolver el siguiente problema de implementación. Un paquete de implementación almacenado en una ubicación de Amazon S3 se sustituye por un nuevo paquete de implementación que contiene el código de función de Lambda actualizado, pero la propiedad `CodeUri` permanece sin alterar (a diferencia de cuando el nuevo paquete de implementación se carga en una nueva ubicación de Amazon S3 y se cambia `CodeUri` a la nueva ubicación).

Este problema se debe a una AWS SAM plantilla que tiene las siguientes características:

- El objeto `DeploymentPreference` está configurado para despliegues graduales (como se describe en [Implementación gradual de aplicaciones sin servidor con AWS SAM](#))

- La propiedad `AutoPublishAlias` está establecida y no cambia entre despliegues
- La propiedad `CodeUri` está establecida y no cambia entre despliegues.

En este escenario, la actualización de `AutoPublishCodeSha256` da como resultado la creación correcta de una nueva versión de Lambda. Sin embargo, no se reconocerá el código de función nuevo implementado en Amazon S3. Para reconocer el código de función nuevo, considere la posibilidad de utilizar el control de versiones en su bucket de Amazon S3. Especifica la propiedad `Version` de la función de Lambda y configure el bucket para que utilice siempre el paquete de implementación más reciente.

En este escenario, para activar correctamente la implementación gradual, debe proporcionar un valor único para `AutoPublishCodeSha256`.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

`CapacityProviderConfig`

Configura el proveedor de capacidad al que se adjuntarán las versiones publicadas de la función. Esto permite que la función se ejecute en instancias EC2 propiedad del cliente administradas por Lambda Managed Instances.

Tipo: [`CapacityProviderConfig`](#)

Obligatorio: no

CloudFormation compatibilidad: SAM aplanar la propiedad transferida a la propiedad de un `AWS::Lambda::Function` recurso y [`CapacityProviderConfig`](#) reconstruye la estructura anidada.

`CodeSigningConfigArn`

El ARN del recurso [`AWS::Lambda::CodeSigningConfig`](#), que se utiliza para habilitar la firma de código para esta función. Para obtener más información sobre la firma de código, consulta [Configura la firma de código para tu AWS SAM aplicación](#).

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la propiedad de un recurso. [CodeSigningConfigArn](#) `AWS::Lambda::Function`

CodeUri

Código de la función. Los valores aceptados son:

- El URI de Amazon S3 de la función. Por ejemplo, `s3://bucket-123456789/sam-app/1234567890abcdefg`.
- La ruta local a la función. Por ejemplo, `hello_world/`.
- Un objeto [FunctionCode](#).

 Note

Si proporcionas el URI o el objeto [FunctionCode](#) de Amazon S3 de una función, debe hacer referencia a un [paquete de implementación de Lambda](#) válido.

Si proporcionas una ruta de archivo local, usa AWS SAM CLI para cargar el archivo local en el momento de la implementación. Para obtener más información, consulte [Cómo AWS SAM carga archivos locales en el momento de la implementación](#).

Si utiliza funciones intrínsecas en la `CodeUri` propiedad, no AWS SAM podrá analizar correctamente los valores. Considere usar [AWS::LanguageExtensions transformer](#) en su lugar.

Tipo: [Cadena | [FunctionCode](#)]

Obligatorio: condicional. Cuando `PackageType` se establece en `Zip`, se requiere `CodeUri` o `InlineCode`.

CloudFormation compatibilidad: esta propiedad es similar a la [Code](#) propiedad de un `AWS::Lambda::Function` recurso. Las propiedades anidadas de Amazon S3 tienen un nombre diferente.

DeadLetterQueue

Configura un tema de Amazon Simple Notification Service (Amazon SNS) o una cola de Amazon Simple Queue Service (Amazon SQS) donde Lambda envía eventos que no puede procesar. Para obtener más información sobre la funcionalidad de las colas con letra muerta, consulte las [Dead-letter colas](#) en la Guía para desarrolladores.AWS Lambda

Note

Si el origen de eventos de la función de Lambda es una cola de Amazon SQS, configure una cola de mensajes fallidos para la cola de origen, no para la función de Lambda. La cola de mensajes fallidos que configure para una función se utiliza para la [cola de invocación asíncrona](#) de la función, no para las colas de origen de eventos.

Tipo: Mapa | [DeadLetterQueue](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es similar a la [DeadLetterConfig](#) propiedad de un `AWS::Lambda::Function` recurso. En CloudFormation el tipo se deriva de `TargetArn`, mientras que en se AWS SAM debe pasar el tipo junto con el `TargetArn`.

DeploymentPreference

La configuración para habilitar las implementaciones graduales de Lambda.

Si se especifica un `DeploymentPreference` objeto, AWS SAM crea un objeto [AWS::CodeDeploy::Application](#) llamado `ServerlessDeploymentApplication` (uno por pila), un [AWS::CodeDeploy::DeploymentGroup](#) llamado `<function-logical-id>DeploymentGroup` y un [AWS::IAM::Role](#) llamado `CodeDeployServiceRole`.

Tipo: [DeploymentPreference](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

Consulte también: Para obtener más información acerca de esta propiedad, consulta [Implementación gradual de aplicaciones sin servidor con AWS SAM](#).

Description

Una descripción de la función.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Description](#) propiedad de un `AWS::Lambda::Function` recurso.

DurableConfig

Configuración para funciones duraderas. Permite una ejecución continua con funciones automáticas de puntos de control y reproducción.

Tipo: [DurableConfig](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un equivalente. CloudFormation

Environment

La configuración de registro para el entorno de tiempo de ejecución.

Tipo: [entorno](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Environment](#) propiedad de un `AWS::Lambda::Function` recurso.

EphemeralStorage

Objeto que especifica el espacio en disco, en MB, disponible para la función de Lambda en `/tmp`.

Para obtener más información sobre esta propiedad, consulta [Entorno de ejecución de Lambda](#) en la Guía para desarrolladores de AWS Lambda .

Tipo: [EphemeralStorage](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [EphemeralStorage](#) propiedad de un `AWS::Lambda::Function` recurso.

EventInvokeConfig

El objeto que describe la configuración de invocación de eventos en una función de Lambda.

Tipo: [EventInvokeConfiguration](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

Events

Especifica los eventos que activan esta función. Los eventos constan de un tipo y un conjunto de propiedades que dependen del tipo.

Tipo: [EventSource](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

FileSystemConfigs

Lista de [FileSystemConfig](#) objetos que especifican la configuración de conexión de un sistema de archivos Amazon Elastic File System (Amazon EFS) o Amazon S3 Files. Puede conectar un punto de acceso Amazon EFS o un punto de acceso S3 Files, pero no ambos.

Cada `FileSystemConfig` objeto contiene un `Arn` (el ARN del punto de acceso) y una `LocalMountPath` (la ruta en la que se monta el sistema de archivos en la función). En el caso de Amazon EFS, el ARN es un ARN de punto de acceso de Amazon EFS. Para los archivos S3, el ARN es un ARN. `AWS::S3Files::AccessPoint`

Si la plantilla contiene un recurso [AWS::EFS::MountTarget](#), también debe especificar un atributo de recurso `DependsOn` para asegurarse de que el destino de montaje se crea o actualiza antes de la función. Del mismo modo, si la plantilla contiene un `AWS::S3Files::MountTarget` recurso, debe especificar un `DependsOn` atributo para el destino de montaje de S3 Files.

Tipo: lista

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [FileSystemConfigs](#) propiedad de un `AWS::Lambda::Function` recurso.

FunctionName

Un nombre para la función. Si no especificas un nombre, se genera un nombre único.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [FunctionName](#) propiedad de un `AWS::Lambda::Function` recurso.

FunctionScalingConfig

Configura el comportamiento de escalado de las funciones Lambda que se ejecutan en los proveedores de capacidad. Define el número mínimo y máximo de entornos de ejecución.

Tipo: [FunctionScalingConfig](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [FunctionScalingConfig](#) propiedad de un `AWS::Lambda::Function` recurso.

FunctionUrlConfig

El objeto que describe una URL de función. Una URL de función es un punto de conexión HTTPS que puede utilizar para invocar la función.

Para obtener más información, consulta [URL de función](#) en la Guía para desarrolladores de AWS Lambda .

Tipo: [FunctionUrlConfig](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

Handler

La función del código a la que se llama para iniciar la ejecución. Esta propiedad solo es necesaria si la propiedad `PackageType` está establecida en `Zip`.

Tipo: cadena

Obligatorio: condicional

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Handler](#) propiedad de un `AWS::Lambda::Function` recurso.

ImageConfig

El objeto utilizado para configurar los ajustes de la imagen del contenedor Lambda. Para obtener más información, consulta [Usar imágenes de contenedor con Lambda](#) en la Guía para desarrolladores de AWS Lambda .

Tipo: [ImageConfig](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [ImageConfig](#) propiedad de un `AWS::Lambda::Function` recurso.

ImageUri

El URI del repositorio Amazon Elastic Container Registry (Amazon ECR) para la imagen del contenedor de la función de Lambda. Esta propiedad solo se aplica si la propiedad `PackageType` está establecida en `Image`; de lo contrario, se ignora. Para obtener más información, consulta [Usar imágenes de contenedor con Lambda](#) en la Guía para desarrolladores de AWS Lambda .

Note

Si la `PackageType` propiedad está establecida en `Image`, entonces `ImageUri` es obligatoria o bien debe crear la aplicación con `Metadata` las entradas necesarias en el archivo AWS SAM de plantilla. Para obtener más información, consulte [Compilación predeterminada con AWS SAM](#).

La creación de la aplicación con las entradas `Metadata` necesarias tiene prioridad sobre `ImageUri`, por lo que si especifica ambas, `ImageUri` se ignorará.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [ImageUri](#) propiedad del tipo de `AWS::Lambda::Function` Code datos.

InlineCode

El código de la función de Lambda que se escribe directamente en la plantilla. Esta propiedad solo se aplica si la propiedad `PackageType` está establecida en `Zip`; de lo contrario, se ignora.

 Note

Si la propiedad `PackageType` está establecida en `Zip` (por defecto), se requiere `CodeUri` o `InlineCode`.

Tipo: cadena

Obligatorio: condicional

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [ZipFile](#) propiedad del tipo de `AWS::Lambda::Function` datos.

KmsKeyArn

El ARN de una clave AWS Key Management Service (AWS KMS) que Lambda utiliza para cifrar y descifrar las variables de entorno de la función.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [KmsKeyArn](#) propiedad de un recurso. `AWS::Lambda::Function`

Layers

La lista de ARN `LayerVersion` que debe usar esta función. El orden especificado aquí es el orden en el que se importarán al ejecutar la función de Lambda. La versión es un ARN completo que incluye la versión o una referencia a un `LayerVersion` recurso. Por ejemplo, una referencia a `LayerVersion` será `!Ref MyLayer`, mientras que un ARN completo, con la versión incluida, será `arn:aws:lambda:region:account-id:layer:layer-name:version`.

Tipo: lista

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Layers](#) propiedad de un `AWS::Lambda::Function` recurso.

LoggingConfig

Los ajustes de configuración de Amazon CloudWatch Logs de la función.

Tipo: [LoggingConfig](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [LoggingConfig](#) propiedad de un `AWS::Lambda::Function` recurso.

MemorySize

El tamaño de la memoria en MB asignado por cada invocación de la función.

Tipo: entero

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [MemorySize](#) propiedad de un `AWS::Lambda::Function` recurso.

PackageType

El tipo de paquete de implementación de la función de Lambda. Para obtener más información, consulta [Paquetes de implementación de Lambda](#) en la Guía para desarrolladores de AWS Lambda .

Notas:

1. Si esta propiedad se establece en Zip (predeterminado), se aplica `CodeUri` o `InlineCode` y se ignora `ImageUri`.
2. Si esta propiedad está establecida en Image, solo se aplica `ImageUri` y se ignoran `CodeUri` y `InlineCode`. El repositorio ECR de Amazon necesario para almacenar la imagen del contenedor de la función se puede crear automáticamente mediante. AWS SAMCLI Para obtener más información, consulta [sam deploy](#).

Valores válidos: Zip o Image

Tipo: cadena

Obligatorio: no

Valor predeterminado: Zip

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [PackageType](#) propiedad de un `AWS::Lambda::Function` recurso.

PermissionsBoundary

El ARN de un límite de permisos que se utilizará para la función de ejecución de esta función. Esta propiedad solo funciona si el rol se genera para usted.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [PermissionsBoundary](#) propiedad de un `AWS::IAM::Role` recurso.

Policies

Políticas de permisos para esta función. Las políticas se añadirán a la función de ejecución predeterminada de la función AWS Identity and Access Management (IAM).

Esta propiedad acepta un valor único o una lista de valores. Entre los valores permitidos se incluyen:

- [AWS SAM Plantillas de política de.](#)
- La ARN de una [política administrada por AWS](#) o una [política administrada por el cliente](#).
- [El nombre de una política AWS gestionada de la siguiente lista.](#)
- Una [política de IAM en línea](#) con formato en YAML como un mapa.

Note

Si especifica la propiedad `Role`, pasa por alto esta propiedad.

Tipo: Cadena | Lista | Mapa

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es similar a la [Policies](#) propiedad de un `AWS::IAM::Role` recurso.

PublishToLatestPublished

Especifica si se debe publicar la última versión de la función cuando se actualice la función.

Tipo: Booleano

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [PublishToLatestPublished](#) propiedad de un `AWS::Lambda::Function` recurso.

PropagateTags

Indique si quiere pasar o no las etiquetas de la propiedad Tags a los recursos generados [AWS::Serverless::Function](#). Especifica `True` para propagar las etiquetas en los recursos generados.

Tipo: Booleano

Obligatorio: no

Valor predeterminado: `False`

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

ProvisionedConcurrencyConfig

La configuración de simultaneidad aprovisionada del alias de una función.

Note

`ProvisionedConcurrencyConfig` puede especificarse solamente si se estableció `AutoPublishAlias`. De lo contrario, se producirá un error.

Tipo: [ProvisionedConcurrencyConfig](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [ProvisionedConcurrencyConfig](#) propiedad de un `AWS::Lambda::Alias` recurso.

RecursiveLoop

El estado de la configuración de detección de bucles recursivos de su función.

Cuando este valor se establece en `Allow` y Lambda detecta que la función se está invocando como parte de un bucle recursivo, no realiza ninguna acción.

Cuando este valor se establece en `Terminate` y Lambda detecta que la función se está invocando como parte de un bucle recursivo, detiene la invocación y se lo notifica.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [RecursiveLoop](#) propiedad del `AWS::Lambda::Function` recurso.

ReservedConcurrentExecutions

El número máximo de ejecuciones simultáneas que quiere reservar para la función.

Para obtener más información, consulta [Escala de funciones de Lambda](#) en la AWS Lambda Guía para desarrolladores de .

Tipo: entero

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [ReservedConcurrentExecutions](#) propiedad de un `AWS::Lambda::Function` recurso.

Role

El ARN de un rol de IAM que se utilizará como rol de ejecución de esta función.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es similar a la [Role](#) propiedad de un `AWS::Lambda::Function` recurso. Esto es obligatorio en CloudFormation , pero no en AWS SAM. Si no se especifica un rol, se crea uno para usted con un identificador lógico de `<function-logical-id>Role`.

RolePath

La ruta a la función de ejecución de IAM de la función.

Utilice esta propiedad cuando se genere el rol para usted. No la utilice si el rol se especifica con la propiedad `Role`.

Tipo: cadena

Obligatorio: condicional

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Path](#) propiedad de un `AWS::IAM::Role` recurso.

Runtime

El identificador del [tiempo de ejecución](#) de la función. Esta propiedad solo es necesaria si la propiedad `PackageType` está establecida en `Zip`.

Note

Si especifica el `provided` identificador de esta propiedad, puede usar el atributo `Metadata resource` para indicar AWS SAM que se cree el tiempo de ejecución personalizado que requiere esta función. Para obtener más información acerca de la creación de tiempos de ejecución personalizados, consulta [Creación de funciones de eventos de Lambda con tiempos de ejecución personalizados en AWS SAM](#).

Tipo: cadena

Obligatorio: condicional

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Runtime](#) propiedad de un `AWS::Lambda::Function` recurso.

RuntimeManagementConfig

Configure las opciones de administración del tiempo de ejecución para las funciones de Lambda, como las actualizaciones del entorno de ejecución, el comportamiento de reversión y la selección de una versión de tiempo de ejecución específica. Para obtener más información, consulta las [actualizaciones del tiempo de ejecución de Lambda](#) en la Guía para desarrolladores de AWS Lambda .

Tipo: [RuntimeManagementConfig](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [RuntimeManagementConfig](#) propiedad de un `AWS::Lambda::Function` recurso.

SnapStart

Cree una instantánea de cualquier versión nueva de la función de Lambda. Una instantánea es un estado en caché de la función inicializada, incluidas todas sus dependencias. La función se

inicializa solo una vez y el estado en caché se reutiliza para todas las invocaciones futuras, lo que mejora el rendimiento de la aplicación al reducir el número de veces que se debe inicializar la función. Para obtener más información, consulte [Mejorar el rendimiento de las empresas emergentes con Lambda SnapStart](#) en la Guía AWS Lambda para desarrolladores.

Tipo: [SnapStart](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [SnapStart](#) propiedad de un `AWS::Lambda::Function` recurso.

SourceKMSKeyArn

Representa un ARN de clave de KMS que se utiliza para cifrar el código de función ZIP del cliente.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [SourceKMSKeyArn](#) propiedad de un tipo de `AWS::Lambda::Function` Code datos.

Tags

Un mapa (cadena a cadena) que especifica las etiquetas agregadas a esta función. Para obtener más información sobre las claves y los valores válidos de las etiquetas, consulta los [requisitos de clave y valor de las etiquetas](#) en la Guía para desarrolladores de AWS Lambda .

Cuando se crea la pila, agrega AWS SAM automáticamente una `lambda:createdBy: SAM` etiqueta a esta función de Lambda y a las funciones predeterminadas que se generan para esta función.

Tipo: mapa

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es similar a la [Tags](#) propiedad de un `AWS::Lambda::Function` recurso. La Tags propiedad en se AWS SAM compone de pares clave-valor (mientras que en CloudFormation esta propiedad se compone de una lista de Tag objetos). Además, agrega AWS SAM automáticamente una `lambda:createdBy: SAM` etiqueta a esta función Lambda y a las funciones predeterminadas que se generan para esta función.

TenancyConfig

Configuración para el modo de aislamiento de inquilinos de Lambda. Garantiza que los entornos de ejecución nunca se compartan entre distintos identificadores de arrendatarios, lo que proporciona un aislamiento a nivel de cómputo para las aplicaciones con varios arrendatarios.

Tipo: [TenancyConfig](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [TenancyConfig](#) propiedad de un recurso. `AWS::Lambda::Function`

Timeout

El tiempo máximo en segundos durante el cual la función se puede poner en marcha.

Tipo: entero

Obligatorio: no

Predeterminado: 3

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Timeout](#) propiedad de un `AWS::Lambda::Function` recurso.

Tracing

La cadena que especifica el modo de X-Ray rastreo de la función.

- `Active`— Activa el X-Ray rastreo de la función.
- `Disabled`— Se desactiva X-Ray para la función.
- `PassThrough`— Activa el X-Ray rastreo de la función. La decisión de muestreo se delega en los servicios intermedios.

Si se especifica como `Active` o `PassThrough` y la propiedad `Role` no está establecida, AWS SAM agrega la política de `arn:aws:iam::aws:policy/AWSXrayWriteOnlyAccess` al rol de ejecución de Lambda que crea para usted.

Para obtener más información al respecto X-Ray, consulte [Utilización AWS Lambda con AWS X-Ray](#) en la Guía para AWS Lambda desarrolladores.

Valores válidos: `[Active|Disabled|PassThrough]`

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es similar a la [TracingConfig](#) propiedad de un `AWS::Lambda::Function` recurso.

VersionDescription

Especifica el campo `Description` que se agrega al recurso de la nueva versión de Lambda.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Description](#) propiedad de un `AWS::Lambda::Version` recurso.

VersionDeletionPolicy

Especifica la política de eliminación del recurso de la versión Lambda que se crea cuando `AutoPublishAlias` se establece. Esto controla si el recurso de versión se conserva o se elimina al eliminar la pila.

Valores válidos: `Delete`, `Retain` o `Snapshot`

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente. Establece el `DeletionPolicy` atributo del `AWS::Lambda::Version` recurso generado.

VpcConfig

La configuración que permite a esta función acceder a los recursos privados dentro de su nube privada virtual (VPC).

Tipo: [VpcConfig](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [VpcConfig](#) propiedad de un `AWS::Lambda::Function` recurso.

valores devueltos

Ref.

Cuando se proporciona el ID lógico de este recurso a la función intrínseca Ref, devuelve el nombre del recurso de la función de Lambda subyacente.

Para obtener más información sobre el uso de la función Ref, consulta [Ref](#) en la Guía del usuario de AWS CloudFormation .

Ventilador:: GetAtt

Fn::GetAtt devuelve un valor para un atributo especificado de este tipo. A continuación se indican los atributos disponibles y los valores devueltos de muestra.

Para obtener más información acerca de cómo usar Fn::GetAtt, consulta [Fn::GetAtt](#) en la Guía del usuario de AWS CloudFormation .

Arn

El ARN de la función de Lambda subyacente.

Ejemplos

Función sencilla

El siguiente es un ejemplo básico de un recurso [AWS::Serverless::Función](#) de tipo de paquete Zip (predeterminado) y código de función en un bucket de Amazon S3.

YAML

```
Type: AWS::Serverless::Function
Properties:
  Handler: index.handler
  Runtime: python3.9
  CodeUri: s3://bucket-name/key-name
```

Ejemplo de propiedades de funciones

El siguiente es un ejemplo de un tipo [AWS::Serverless::Función](#) de paquete Zip (predeterminado) que usa InlineCode,, Layers TracingPolicies, un sistema de Amazon EFS archivos y una fuente de Api eventos.

YAML

```

Type: AWS::Serverless::Function
DependsOn: MyMountTarget          # This is needed if an AWS::EFS::MountTarget resource
                                   is declared for EFS
Properties:
  Handler: index.handler
  Runtime: python3.9
  InlineCode: |
    def handler(event, context):
      print("Hello, world!")
  ReservedConcurrentExecutions: 30
  Layers:
    - Ref: MyLayer
  Tracing: Active
  Timeout: 120
  FileSystemConfigs:
    - Arn: !Ref MyEfsFileSystem
      LocalMountPath: /mnt/EFS
  Policies:
    - AWSLambdaExecute
    - Version: '2012-10-17'
      Statement:
        - Effect: Allow
          Action:
            - s3:GetObject
            - s3:GetObjectACL
          Resource: 'arn:aws:s3:::sam-s3-demo-bucket/*'
  Events:
    ApiEvent:
      Type: Api
      Properties:
        Path: /path
        Method: get

```

ImageConfig ejemplo

A continuación, se muestra un ejemplo de una ImageConfig para una función de Lambda de tipo paquete Image.

YAML

```

HelloWorldFunction:

```

```
Type: AWS::Serverless::Function
Properties:
  PackageType: Image
  ImageUri: account-id.dkr.ecr.region.amazonaws.com/ecr-repo-name:image-name
  ImageConfig:
    Command:
      - "app.lambda_handler"
    EntryPoint:
      - "entrypoint1"
    WorkingDirectory: "workDir"
```

RuntimeManagementConfig ejemplos

Una función de Lambda configurada para actualizar su tiempo de ejecución de ejecución de acuerdo con el comportamiento actual:

```
TestFunction
Type: AWS::Serverless::Function
Properties:
  ...
  Runtime: python3.9
  RuntimeManagementConfig:
    UpdateRuntimeOn: Auto
```

Una función de Lambda configurada para actualizar su entorno de tiempo de ejecución cuando se actualiza la función:

```
TestFunction
Type: AWS::Serverless::Function
Properties:
  ...
  Runtime: python3.9
  RuntimeManagementConfig:
    UpdateRuntimeOn: FunctionUpdate
```

Una función de Lambda configurada para actualizar su entorno de tiempo de ejecución de forma manual:

```
TestFunction
Type: AWS::Serverless::Function
Properties:
  ...
```



```
Runtime: python3.9
RuntimeManagementConfig:
  RuntimeVersionArn: arn:aws:lambda:us-
east-1::runtime:4c459dd0104ee29ec65dcad056c0b3ddb20d6db76b265ade7eda9a066859b1e
  UpdateRuntimeOn: Manual
```

SnapStart ejemplos

Ejemplo de una función Lambda SnapStart activada para futuras versiones:

```
TestFunc
Type: AWS::Serverless::Function
Properties:
  ...
  SnapStart:
    ApplyOn: PublishedVersions
```

TenancyConfig ejemplos

Ejemplo de una función Lambda con el modo de aislamiento de inquilinos activado:

```
TestFunction
Type: AWS::Serverless::Function
Properties:
  ...
  TenancyConfig:
    TenantIsolationMode: PER_TENANT
```

Ejemplo de sistema de archivos S3 Files

El siguiente ejemplo crea una función Lambda que monta un sistema de archivos Amazon S3 Files. La plantilla crea un depósito de S3, un sistema de archivos de S3 Files respaldado por ese depósito, un destino de montaje en una subred de VPC y un punto de acceso. La función monta el punto de acceso en el depósito de `/mnt/s3files` S3 y puede leer y escribir archivos que se sincronizan con él.

Note

Puede adjuntar un sistema de archivos Amazon EFS o un sistema de archivos S3 Files a una función Lambda, pero no ambos al mismo tiempo.

YAML

```
AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31
Description: Lambda function with S3 Files file system

Resources:
  # VPC and networking
  MyVpc:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: 10.0.0.0/16
      EnableDnsSupport: true
      EnableDnsHostnames: true

  MySubnet:
    Type: AWS::EC2::Subnet
    Properties:
      VpcId: !Ref MyVpc
      CidrBlock: 10.0.1.0/24
      AvailabilityZone: !Select [0, !GetAZs '']

  MySecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupDescription: Security group for Lambda and S3 Files
      VpcId: !Ref MyVpc
      SecurityGroupIngress:
        - IpProtocol: tcp
          FromPort: 2049
          ToPort: 2049
          CidrIp: 10.0.0.0/16

  # S3 bucket for file storage
  MyS3Bucket:
    Type: AWS::S3::Bucket

  # IAM role for S3 Files to access the bucket
  S3FilesRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Version: '2012-10-17'
        Statement:
```

```

    - Effect: Allow
    Principal:
      Service: elasticfilesystem.amazonaws.com
    Action: sts:AssumeRole
Policies:
  - PolicyName: S3FilesBucketAccess
    PolicyDocument:
      Version: '2012-10-17'
      Statement:
        - Effect: Allow
          Action:
            - s3:GetObject
            - s3:PutObject
            - s3:DeleteObject
            - s3:ListBucket
          Resource:
            - !GetAtt MyS3Bucket.Arn
            - !Sub '${MyS3Bucket.Arn}/*'

```

```
# S3 Files resources
```

```
MyS3FilesFileSystem:
```

```
  Type: AWS::S3Files::FileSystem
```

```
  Properties:
```

```
    Bucket: !GetAtt MyS3Bucket.Arn
```

```
    RoleArn: !GetAtt S3FilesRole.Arn
```

```
MyS3FilesMountTarget:
```

```
  Type: AWS::S3Files::MountTarget
```

```
  Properties:
```

```
    FileSystemId: !Ref MyS3FilesFileSystem
```

```
    SubnetId: !Ref MySubnet
```

```
    SecurityGroups:
```

```
      - !Ref MySecurityGroup
```

```
MyS3FilesAccessPoint:
```

```
  Type: AWS::S3Files::AccessPoint
```

```
  Properties:
```

```
    FileSystemId: !Ref MyS3FilesFileSystem
```

```
    PosixUser:
```

```
      Uid: '1000'
```

```
      Gid: '1000'
```

```
    RootDirectory:
```

```
      Path: /lambda
```

```
    CreationInfo:
```

```
    OwnerUid: '1000'
    OwnerGid: '1000'
    Permissions: '750'

# Lambda function with S3 Files mount
MyFunction:
  Type: AWS::Serverless::Function
  DependsOn: MyS3FilesMountTarget
  Properties:
    Handler: index.handler
    Runtime: python3.12
    Timeout: 120
    VpcConfig:
      SecurityGroupIds:
        - !Ref MySecurityGroup
      SubnetIds:
        - !Ref MySubnet
    FileSystemConfigs:
      - Arn: !GetAtt MyS3FilesAccessPoint.AccessPointArn
        LocalMountPath: /mnt/s3files
    Policies:
      - Version: '2012-10-17'
        Statement:
          - Effect: Allow
            Action:
              - s3files:ClientMount
              - s3files:ClientWrite
            Resource: !GetAtt MyS3FilesAccessPoint.AccessPointArn
  InlineCode: |
    import os

    def handler(event, context):
      # Write a file to the S3 Files mount
      with open('/mnt/s3files/hello.txt', 'w') as f:
        f.write('Hello from Lambda!')

      # List files at the mount path
      files = os.listdir('/mnt/s3files')
      return {'files': files}
```

DeadLetterQueue

Especifica una cola de SQS o un tema de SNS al que (AWS Lambda Lambda) envía eventos cuando no puede procesarlos. [Para obtener más información sobre la funcionalidad de las colas de cartas muertas, consulte Dead-letter las colas en la Guía para desarrolladores.AWS Lambda](#)

SAM añadirá automáticamente los permisos adecuados a su función de ejecución de la función de eventos de Lambda para permitir que el servicio Lambda acceda al recurso. sqs: se SendMessage añadirá para las colas de SQS y SNS:Publish para los temas de SNS.

Sintaxis

Para declarar esta entidad en su plantilla (), utilice la siguiente sintaxis. AWS Serverless Application Model AWS SAM

YAML

```
TargetArn: String  
Type: String
```

Propiedades

TargetArn

El Nombre de recurso de Amazon (ARN) de una cola de Amazon SQS o del tema de Amazon SNS.

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [TargetArn](#) propiedad del tipo de AWS::Lambda::Function DeadLetterConfig datos.

Type

Tipo de cola de mensajes fallidos.

Valores válidos: SNS, SQS

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

Ejemplos

DeadLetterQueue

Ejemplo de cola de mensaje fallido para un tema de SNS.

YAML

```
DeadLetterQueue:
  Type: SNS
  TargetArn: arn:aws:sns:us-east-2:123456789012:my-topic
```

DeploymentPreference

Especifica las configuraciones para permitir las implementaciones graduales de Lambda. Para obtener más información acerca de la configuración de implementaciones graduales de Lambda, consulte [Implementación gradual de aplicaciones sin servidor con AWS SAM](#).

Note

Debe especificar un `AutoPublishAlias` en su objeto [AWS::Serverless::Función](#) para utilizar un objeto `DeploymentPreference`; de lo contrario, se producirá un error.

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
Alarms: List
Enabled: Boolean
Hooks: Hooks
PassthroughCondition: Boolean
Role: String
TriggerConfigurations: List
```

Type: *String*

Propiedades

Alarms

Una lista de CloudWatch las alarmas que desea que se activen debido a cualquier error provocado por la implementación.

Esta propiedad acepta la función intrínseca Fn: :If. Consulte la sección de ejemplos al final de este tema para ver una plantilla de ejemplo que utiliza Fn: :If.

Tipo: lista

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

Enabled

Si esta preferencia de implementación está habilitada.

Tipo: Booleano

Obligatorio: no

Valor predeterminado: True

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

Hooks

Funciones de eventos Lambda de validación que se ejecutan antes y después del cambio de tráfico.

Tipo: [enlaces](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

PassthroughCondition

Si es True y esta preferencia de despliegue está habilitada, la condición de la función se transferirá al CodeDeploy recurso generado. Por lo general, debe establecerlo en True. De lo contrario, el CodeDeploy recurso se crearía incluso si la condición de la función se resolviera como False.

Tipo: Booleano

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

Role

Un ARN de rol de IAM CodeDeploy que se utilizará para el cambio de tráfico. No se creará una función de rol de IAM si se proporciona.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

TriggerConfigurations

Una lista de las configuraciones de activación que desea asociar al grupo de implementación. Se usa para notificar a un tema de SNS sobre eventos del ciclo de vida.

Tipo: lista

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [TriggerConfigurations](#) propiedad de un `AWS::CodeDeploy::DeploymentGroup` recurso.

Type

Actualmente, hay dos categorías de tipos de implementación: Linear (lineal) y Canary (valor controlado). Para obtener más información acerca de los tipos de implementaciones disponibles, consulte [Implementación gradual de aplicaciones sin servidor con AWS SAM](#).

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

Ejemplos

DeploymentPreference con ganchos antes y después del tráfico.

Ejemplo de preferencia de implementación que contiene enlaces previos y posteriores al tráfico.

YAML

```
DeploymentPreference:
  Enabled: true
  Type: Canary10Percent10Minutes
  Alarms:
    - !Ref: AliasErrorMetricGreaterThanZeroAlarm
    - !Ref: LatestVersionErrorMetricGreaterThanZeroAlarm
  Hooks:
    PreTraffic:
      !Ref: PreTrafficLambdaFunction
    PostTraffic:
      !Ref: PostTrafficLambdaFunction
```

DeploymentPreference con Fn::If es una función intrínseca

Ejemplo de preferencia de implementación que se utiliza para configurar las alarmas Fn::If. En este ejemplo, Alarm1 se configurará si MyCondition es true y Alarm2 y Alarm5 configurarán si MyCondition es false.

YAML

```
DeploymentPreference:
  Enabled: true
  Type: Canary10Percent10Minutes
  Alarms:
    Fn::If:
      - MyCondition
      - - Alarm1
        - Alarm2
        - Alarm5
```

Enlaces

Funciones de eventos Lambda de validación que se ejecutan antes y después del cambio de tráfico.

Note

Las funciones de Lambda a las que se hace referencia en esta propiedad configuran el objeto `CodeDeployLambdaAliasUpdate` del recurso resultante [AWS::Lambda::Alias](#). Para obtener más información, consulte la [CodeDeployLambdaAliasUpdate política](#) en la Guía del AWS CloudFormation usuario.

Sintaxis

Para declarar esta entidad en su plantilla AWS Serverless Application Model (AWS SAM), utilice la siguiente sintaxis.

YAML

```
PostTraffic: String  
PreTraffic: String
```

Propiedades

PostTraffic

Función de Lambda que se ejecuta después del desvío del tráfico.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

PreTraffic

Función de Lambda que se ejecuta antes del desvío del tráfico.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

Ejemplos

Enlaces

Ejemplo de funciones de enlace

YAML

```
Hooks:
  PreTraffic:
    Ref: PreTrafficLambdaFunction
  PostTraffic:
    Ref: PostTrafficLambdaFunction
```

DurableConfig

Configura los ajustes de ejecución duraderos de las AWS Lambda funciones. Las funciones duraderas pueden funcionar hasta un año y comprobar automáticamente el progreso, lo que permite ejecutar flujos de trabajo prolongados y aplicaciones tolerantes a fallos. Para obtener más información sobre las funciones duraderas, consulte [Funciones duraderas de Lambda](#) en la Guía para AWS Lambda desarrolladores.

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
ExecutionTimeout: Integer
RetentionPeriodInDays: Integer
```

Propiedades

ExecutionTimeout

La cantidad de tiempo (en segundos) que Lambda permite que una función duradera se ejecute antes de detenerla. El máximo es un año de 366 días o 31.622.400 segundos.

Tipo: entero

Obligatorio: sí

Mínimo: 1

Máximo: 31622400

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [ExecutionTimeout](#) propiedad del tipo de AWS::Lambda::Function DurableConfig datos.

RetentionPeriodInDays

El número de días que Lambda conserva su historial tras el cierre de una ejecución duradera, de uno a 90 días. El valor predeterminado es 14 días.

Tipo: entero

Obligatorio: no

Predeterminado: 14

Mínimo: 1

Máximo: 90

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [RetentionPeriodInDays](#) propiedad del tipo de AWS::Lambda::Function DurableConfig datos.

Ejemplos

DurableConfig

Ejemplo de configuración duradera para una función con un tiempo de espera de ejecución de 1 hora y un período de retención de 7 días.

YAML

```
DurableConfig:
  ExecutionTimeout: 3600
  RetentionPeriodInDays: 7
```

EventInvokeConfiguration

Opciones de configuración para invocaciones [asíncronas](#) de alias o versiones de Lambda.

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
DestinationConfig: EventInvokeDestinationConfiguration  
MaximumEventAgeInSeconds: Integer  
MaximumRetryAttempts: Integer
```

Propiedades

DestinationConfig

Un objeto de configuración que especifica el destino de un evento después de que Lambda lo procese.

Tipo: [EventInvokeDestinationConfiguration](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es similar a la [DestinationConfig](#) propiedad de un `AWS::Lambda::EventInvokeConfig` recurso. SAM requiere un parámetro adicional, «Type», que no existe en CloudFormation.

MaximumEventAgeInSeconds

La antigüedad máxima de una solicitud que Lambda envía a una función para su procesamiento.

Tipo: entero

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [MaximumEventAgeInSeconds](#) propiedad de un `AWS::Lambda::EventInvokeConfig` recurso.

MaximumRetryAttempts

El número máximo de veces que se puede reintentar antes de que la función devuelva un error.

Tipo: entero

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [MaximumRetryAttempts](#) propiedad de un `AWS::Lambda::EventInvokeConfig` recurso.

Ejemplos

MaximumEventAgeInSeconds

MaximumEventAgeInSeconds ejemplo

YAML

```
EventInvokeConfig:
  MaximumEventAgeInSeconds: 60
  MaximumRetryAttempts: 2
  DestinationConfig:
    OnSuccess:
      Type: SQS
      Destination: arn:aws:sqs:us-west-2:012345678901:my-queue
    OnFailure:
      Type: Lambda
      Destination: !GetAtt DestinationLambda.Arn
```

EventInvokeDestinationConfiguration

Un objeto de configuración que especifica el destino de un evento después de que Lambda lo procese.

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
OnFailure: OnFailure
```

[OnSuccess](#): [OnSuccess](#)

Propiedades

OnFailure

Un destino para eventos que no han podido procesarse.

Tipo: [OnFailure](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es similar a la [OnFailure](#) propiedad de un `AWS::Lambda::EventInvokeConfig` recurso. Requiere `Type`, una propiedad adicional exclusiva para SAM.

OnSuccess

Un destino para eventos que se procesaron correctamente.

Tipo: [OnSuccess](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es similar a la [OnSuccess](#) propiedad de un `AWS::Lambda::EventInvokeConfig` recurso. Requiere `Type`, una propiedad adicional exclusiva para SAM.

Ejemplos

OnSuccess

OnSuccess ejemplo

YAML

```
EventInvokeConfig:
  DestinationConfig:
    OnSuccess:
      Type: SQS
      Destination: arn:aws:sqs:us-west-2:012345678901:my-queue
    OnFailure:
      Type: Lambda
```

```
Destination: !GetAtt DestinationLambda.Arn
```

OnFailure

Un destino para eventos que no han podido procesarse.

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
Destination: String  
Type: String
```

Propiedades

Destination

El nombre de recurso de Amazon (ARN) del recurso de destino.

Tipo: cadena

Obligatorio: condicional

CloudFormation compatibilidad: esta propiedad es similar a la [OnFailure](#) propiedad de un `AWS::Lambda::EventInvokeConfig` recurso. SAM añadirá los permisos necesarios al rol de IAM generado automáticamente asociado a esta función para acceder al recurso al que se hace referencia en esta propiedad.

Notas adicionales: Si el tipo es `Lambda/EventBridge`, se requiere el destino.

Type

Tipo de recurso al que se hace referencia en el destino. Los tipos admitidos son `SQS`, `SNS`, `S3`, `Lambda` y `EventBridge`.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

Notas adicionales: Si el tipo es SQS/SNS y la Destination propiedad se deja en blanco, SAM genera automáticamente el SQS/SNS recurso. Para hacer referencia al recurso, utilice `<function-logical-id>.DestinationQueue` para SQS o `<function-logical-id>.DestinationTopic` para SNS. Si el tipo es Lambda/EventBridge, Destination es obligatorio.

Ejemplos

EventInvoke Ejemplo de configuración con destinos SQS y Lambda

En este ejemplo, no se proporciona ningún destino para la onSuccess configuración de SQS, por lo que SAM crea implícitamente una cola de SQS y añade los permisos necesarios. También en este ejemplo, en la onFailure configuración se especifica un destino para un recurso de Lambda declarado en el archivo de plantilla, de modo que SAM añade los permisos necesarios a esta función de eventos de Lambda para llamar a la función de Lambda de destino.

YAML

```
EventInvokeConfig:
  DestinationConfig:
    OnSuccess:
      Type: SQS
    OnFailure:
      Type: Lambda
      Destination: !GetAtt DestinationLambda.Arn # Arn of a Lambda function declared
in the template file.
```

EventInvoke Ejemplo de configuración con destino SNS

En este ejemplo, se proporciona un destino para un tema de SNS declarado en el archivo de plantilla de la onSuccess configuración.

YAML

```
EventInvokeConfig:
  DestinationConfig:
    OnSuccess:
      Type: SNS
      Destination:
        Ref: DestinationSNS # Arn of an SNS topic declared in the tempate file
```

OnSuccess

Un destino para eventos que se procesaron correctamente.

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
Destination: String  
Type: String
```

Propiedades

Destination

El nombre de recurso de Amazon (ARN) del recurso de destino.

Tipo: cadena

Obligatorio: condicional

CloudFormation compatibilidad: esta propiedad es similar a la [OnSuccess](#) propiedad de un `AWS::Lambda::EventInvokeConfig` recurso. SAM añadirá los permisos necesarios al rol de IAM generado automáticamente asociado a esta función para acceder al recurso al que se hace referencia en esta propiedad.

Notas adicionales: Si el tipo es `Lambda/EventBridge`, se requiere el destino.

Type

Tipo de recurso al que se hace referencia en el destino. Los tipos admitidos son `SQS`, `SNS`, `S3`, `Lambda` y `EventBridge`.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

Notas adicionales: Si el tipo es SQS/SNS y la Destination propiedad se deja en blanco, SAM genera automáticamente el SQS/SNS recurso. Para hacer referencia al recurso, utilice `<function-logical-id>.DestinationQueue` para SQS o `<function-logical-id>.DestinationTopic` para SNS. Si el tipo es Lambda/EventBridge, Destination es obligatorio.

Ejemplos

EventInvoke Ejemplo de configuración con destinos SQS y Lambda

En este ejemplo, no se proporciona ningún destino para la onSuccess configuración de SQS, por lo que SAM crea implícitamente una cola de SQS y añade los permisos necesarios. También en este ejemplo, en la onFailure configuración se especifica un destino para un recurso de Lambda declarado en el archivo de plantilla, de modo que SAM añade los permisos necesarios a esta función de eventos de Lambda para llamar a la función de Lambda de destino.

YAML

```
EventInvokeConfig:
  DestinationConfig:
    OnSuccess:
      Type: SQS
    OnFailure:
      Type: Lambda
      Destination: !GetAtt DestinationLambda.Arn # Arn of a Lambda function declared
in the template file.
```

EventInvoke Ejemplo de configuración con destino SNS

En este ejemplo, se proporciona un destino para un tema de SNS declarado en el archivo de plantilla de la onSuccess configuración.

YAML

```
EventInvokeConfig:
  DestinationConfig:
    OnSuccess:
      Type: SNS
      Destination:
        Ref: DestinationSNS # Arn of an SNS topic declared in the tempate file
```

EventSource

El objeto que describe la fuente de los eventos que activan la función. Cada evento consta de un tipo y un conjunto de propiedades que dependen de ese tipo. Para obtener información acerca de las propiedades de cada origen de eventos, consulta el tema correspondiente a ese tipo.

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
Properties: AlexaSkill | Api | CloudWatchEvent | CloudWatchLogs | Cognito
| DocumentDB | DynamoDB | EventBridgeRule | HttpApi | IoTRule | Kinesis | MQ | MSK
| S3 | Schedule | ScheduleV2 | SelfManagedKafka | SNS | SQS
Type: String
```

Propiedades

Properties

Objeto que describe las propiedades de esta asignación de eventos. El conjunto de propiedades debe ajustarse al tipo definido.

Tipo : [AlexaSkill](#) | [Api](#) | [CloudWatchEvent](#) | [Cognito](#) | [CloudWatchLogs](#) | [DocumentDB](#) | [DynamoDB](#) | [Io](#) | [Kinesis](#) | [MQ](#) | [EventBridgeRule](#) | [HttpApi](#) | [MSK](#) | [S3](#) | [TRule](#) | [Programación](#) | [ScheduleV2](#) | [SNS](#) | [SQS](#) | [SelfManagedKafka](#)

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad es exclusiva y no tiene un equivalente. AWS SAM CloudFormation

Type

Tipo de evento.

Valores válidos: `AlexaSkill`, `Api`, `CloudWatchEvent`, `CloudWatchLogs`, `Cognito`, `DocumentDB`, `DynamoDB`, `EventBridgeRule`, `HttpApi`, `IoTRule`, `Kinesis`, `MQ`, `MSK`, `S3`, `Schedule`, `ScheduleV2`, `SelfManagedKafka`, `SNS`, `SQS`

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

Ejemplos

APIEvent

Ejemplo de uso de un evento de API

YAML

```
ApiEvent:
  Type: Api
  Properties:
    Method: get
    Path: /group/{user}
    RestApiId:
      Ref: MyApi
```

AlexaSkill

El objeto que describe un tipo de fuente de evento de AlexaSkill.

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
SkillId: String
```

Propiedades

SkillId

La Alexa Skill ID para la habilidad de Alexa. Para obtener más información sobre la Skill ID, consulta [Configurar el activador de una función de Lambda](#) en la documentación del Alexa Skills Kit.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

Ejemplos

AlexaSkillTrigger

Ejemplo de evento de habilidad de Alexa

YAML

```
AlexaSkillEvent:
  Type: AlexaSkill
```

Api

El objeto que describe un tipo de fuente de evento de Api. Si se define un recurso [AWS::Serverless::API](#), los valores de ruta y método deben corresponder a una operación de la definición de OpenAPI de la API.

Si no se define ningún [AWS::Serverless::API](#), la entrada y la salida de la función son una representación de la solicitud HTTP y la respuesta HTTP.

Por ejemplo, mediante la JavaScript API, el código de estado y el cuerpo de la respuesta se pueden controlar devolviendo un objeto con las claves `StatusCode` y `body`.

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
Auth: ApiFunctionAuth
Method: String
Path: String
```

```
RequestModel: RequestModel  
RequestParameters: List of [ String | RequestParameter ]  
RestApiId: String  
ResponseTransferMode: String  
TimeoutInMillis: Integer
```

Propiedades

Auth

Configuración de autenticación para esta API, ruta y método específicos.

Útil para anular la configuración `DefaultAuthorizer` de autenticación de la API en una ruta individual cuando no se especifica ninguna `DefaultAuthorizer` o la anulación de la configuración `ApiKeyRequired` predeterminada.

Tipo: [ApiFunctionAuth](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

Method

Método HTTP para el que se invoca esta función. Las opciones incluyen DELETE, GET, HEAD, OPTIONS, PATCH, POST, PUT y ANY. Consulte [Configuración de un método HTTP](#) en la Guía para desarrolladores de API Gateway para obtener más información.

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

Path

Ruta de URI para la que se invoca esta función. Debe empezar con `/`.

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

RequestModel

Modelo de solicitud para usarlo en esta API, ruta y método específicos. Debe hacer referencia al nombre de un modelo especificado en la sección `Models` de un recurso [AWS::Serverless::API](#).

Tipo: [RequestModel](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

RequestParameters

Configuración de los parámetro de solicitud para una API+ruta+método específicos. Todos los nombres de los parámetros deben empezar por `method.request` y estar limitados a `method.request.header`, `method.request.querystring` o `method.request.path`.

Una lista puede contener tanto cadenas de nombres de parámetros como [RequestParameter](#) objetos. En el caso de las cadenas, las propiedades `Required` y `Caching` se establecerán de forma predeterminada en `false`.

Tipo: Lista de [Cadena | [RequestParameter](#)]

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

RestApiId

Identificador de un RestApi recurso, que debe contener una operación con la ruta y el método dados. Normalmente, se configura para hacer referencia a un recurso [AWS::Serverless::API](#) definido en esta plantilla.

Si no define esta propiedad, AWS SAM crea un [AWS::Serverless::API](#) recurso predeterminado a partir de un OpenApi documento generado. Ese recurso contiene una unión de todas las rutas y métodos definidos por eventos Api de la misma plantilla que no especifican un RestApiId.

Esto no puede hacer referencia a un recurso [AWS::Serverless::API](#) definido en otra plantilla.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

ResponseTransferMode

El modo de transferencia de respuestas para la integración de la función Lambda. Configúrelo en `RESPONSE_STREAM` para habilitar la transmisión de respuestas de Lambda a través de API Gateway, lo que permite que la función transmita las respuestas a los clientes. Cuando se establece en `RESPONSE_STREAM`, API Gateway utiliza la API Lambda `InvokeWithResponseStreaming`.

Tipo: cadena

Obligatorio: no

Valores válidos: `BUFFERED` | `RESPONSE_STREAM`

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [ResponseTransferMode](#) propiedad de un `AWS::ApiGateway::Method` Integration.

TimeoutInMillis

Tiempo de espera personalizado entre 50 y 29 000 milisegundos.

Note

Al especificar esta propiedad, AWS SAM modifica la definición de OpenAPI. La definición de OpenAPI debe especificarse en línea mediante la propiedad `DefinitionBody`.

Tipo: entero

Obligatorio: no

Valor predeterminado: 29 000 milisegundos (29 segundos)

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

Ejemplos

Ejemplo básico

YAML

```
Events:
  ApiEvent:
    Type: Api
    Properties:
      Path: /path
      Method: get
      RequestParameters:
        - method.request.header.Authorization
        - method.request.querystring.keyword:
            Required: true
            Caching: false
```

ApiFunctionAuth

Configura la autorización a nivel de evento para una API, una ruta y un método específicos.

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
ApiKeyRequired: Boolean
AuthorizationScopes: List
Authorizer: String
InvokeRole: String
OverrideApiAuth: Boolean
ResourcePolicy: ResourcePolicyStatement
```

Propiedades

ApiKeyRequired

Requiere una clave de API para esta API, ruta y método.

Tipo: Booleano

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

AuthorizationScopes

Los ámbitos de autorización que se aplicarán a esta API, ruta y método.

Los ámbitos que especifica anularán los ámbitos aplicados por la propiedad de `DefaultAuthorizer` si los ha especificado.

Tipo: lista

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

Authorizer

`Authorizer` para una función específica.

Si ha especificado un autorizador global para su recurso `AWS::Serverless::Api`, puede anular el autorizador configurando `Authorizer` como `NONE`. Para ver un ejemplo, consulta [Anule un autorizador global para su API de REST de Amazon API Gateway](#).

Note

Si utiliza la propiedad `DefinitionBody` de un recurso de `AWS::Serverless::Api` para describir su API, debe usar `OverrideApiAuth` con `Authorizer` para anular su autorizador global. Para obtener más información, consulta [OverrideApiAuth](#).

Valores válidos: `AWS_IAMNONE`, o el identificador lógico de cualquier autorizador definido en la AWS SAM plantilla.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

InvokeRole

Especifica qué InvokeRole utilizar para la autorización de AWS_IAM.

Tipo: cadena

Obligatorio: no

Valor predeterminado: CALLER_CREDENTIALS

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

Notas adicionales: CALLER_CREDENTIALS asigna a `arn:aws:iam::<user>/`, que utiliza las credenciales de la persona que llama para invocar el punto de conexión.

OverrideApiAuth

Especifica como `true` para anular la configuración del autorizador global de su recurso `AWS::Serverless::Api`. Esta propiedad solo es necesaria si usted especifica un autorizador global y utiliza la propiedad `DefinitionBody` de un recurso `AWS::Serverless::Api` para describir su API.

Note

Si especificas `OverrideApiAuth` como `true`, AWS SAM anulará tu autorizador global con los valores proporcionados para `ApiKeyRequiredAuthorizer`, o. `ResourcePolicy` Por lo tanto, al utilizar `OverrideApiAuth` también se debe especificar al menos una de estas propiedades. Para ver un ejemplo, consulta [Anula un autorizador global cuando se especifica DefinitionBody for AWS::Serverless: :Api](#).

Tipo: Booleano

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

ResourcePolicy

Configura la política de recursos para esta ruta en una API.

Tipo: [ResourcePolicyStatement](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

Ejemplos

Function-Auth

El siguiente ejemplo especifica la autorización a nivel de función.

YAML

```
Auth:
  ApiKeyRequired: true
  Authorizer: NONE
```

Anule un autorizador global para su API de REST de Amazon API Gateway

Puedes especificar un autorizador global para su recurso `AWS::Serverless::Api`. A continuación se muestra un ejemplo que configura un autorizador predeterminado global:

```
AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31
...
Resources:
  MyApiWithLambdaRequestAuth:
    Type: AWS::Serverless::Api
    Properties:
      ...
      Auth:
        Authorizers:
          MyLambdaRequestAuth:
            FunctionArn: !GetAtt MyAuthFn.Arn
            DefaultAuthorizer: MyLambdaRequestAuth
```

Para anular el autorizador predeterminado de su función de AWS Lambda eventos, puede especificar como. Authorizer NONE A continuación, se muestra un ejemplo:

```
AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31
...
Resources:
  ...
  MyFn:
    Type: AWS::Serverless::Function
    Properties:
      ...
      Events:
        LambdaRequest:
          Type: Api
          Properties:
            RestApiId: !Ref MyApiWithLambdaRequestAuth
            Method: GET
            Auth:
              Authorizer: NONE
```

Anula un autorizador global cuando se especifica DefinitionBody for AWS::Serverless: :Api

Al utilizar la propiedad DefinitionBody para describir el recurso AWS::Serverless: :Api, el método de anulación anterior no funciona. A continuación se muestra un ejemplo del uso de la propiedad DefinitionBody para un recurso AWS::Serverless: :Api:

```
AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31
...
Resources:
  MyApiWithLambdaRequestAuth:
    Type: AWS::Serverless::Api
    Properties:
      ...
      DefinitionBody:
        swagger: 2.0
        ...
        paths:
          /lambda-request:
            ...
      Auth:
        Authorizers:
```

```

MyLambdaRequestAuth:
  FunctionArn: !GetAtt MyAuthFn.Arn
  DefaultAuthorizer: MyLambdaRequestAuth

```

Para anular el autorizador global, utilice la propiedad `OverrideApiAuth`. El siguiente es un ejemplo que utiliza `OverrideApiAuth` para anular el autorizador global con el valor proporcionado para `Authorizer`:

```

AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31
...
Resources:
  MyApiWithLambdaRequestAuth:
    Type: AWS::Serverless::Api
    Properties:
      ...
      DefinitionBody:
        swagger: 2-0
        ...
        paths:
          /lambda-request:
            ...
      Auth:
        Authorizers:
          MyLambdaRequestAuth:
            FunctionArn: !GetAtt MyAuthFn.Arn
            DefaultAuthorizer: MyLambdaRequestAuth

  MyAuthFn:
    Type: AWS::Serverless::Function
    ...

  MyFn:
    Type: AWS::Serverless::Function
    Properties:
      ...
      Events:
        LambdaRequest:
          Type: Api
          Properties:
            RestApiId: !Ref MyApiWithLambdaRequestAuth
            Method: GET
            Auth:

```

```
Authorizer: NONE
OverrideApiAuth: true
Path: /lambda-token
```

ResourcePolicyStatement

Configura una política de recursos para todos los métodos y rutas de una API. Para obtener más información sobre las políticas de recursos, consulta [Controlar el acceso a una API con las políticas de recursos de API Gateway](#) en la Guía para desarrolladores de API Gateway.

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
AwsAccountBlacklist: List
AwsAccountWhitelist: List
CustomStatements: List
IntrinsicVpcBlacklist: List
IntrinsicVpcWhitelist: List
IntrinsicVpceBlacklist: List
IntrinsicVpceWhitelist: List
IpRangeBlacklist: List
IpRangeWhitelist: List
SourceVpcBlacklist: List
SourceVpcWhitelist: List
```

Propiedades

AwsAccountBlacklist

Las AWS cuentas que se van a bloquear.

Tipo: lista de cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

AwsAccountWhitelist

Las AWS cuentas que se van a permitir. Para obtener un ejemplo del uso de esta propiedad, consulta la sección Ejemplos al final de esta página.

Tipo: lista de cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

CustomStatements

Una lista de instrucciones de política de recursos personalizadas para aplicarlas a esta API. Para obtener un ejemplo del uso de esta propiedad, consulta la sección Ejemplos al final de esta página.

Tipo: lista

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

IntrinsicVpcBlacklist

La lista de nubes privadas virtuales (VPCs) que se van a bloquear, en la que cada VPC se especifica como referencia, por ejemplo, una [referencia dinámica](#) o la función Ref [intrínseca](#). Para obtener un ejemplo del uso de esta propiedad, consulta la sección Ejemplos al final de esta página.

Tipo: lista

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

IntrinsicVpcWhitelist

La lista de VPCs permitidos, donde cada VPC se especifica como una referencia, como una [referencia dinámica](#) o la función Ref [intrínseca](#).

Tipo: lista

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

`IntrinsicVpceBlacklist`

La lista de puntos de conexión de VPC que se van a bloquear, donde cada punto de conexión de VPC se especifica como referencia, por ejemplo, una [referencia dinámica](#) o la [función intrínseca Ref](#).

Tipo: lista

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

`IntrinsicVpceWhitelist`

La lista de nubes privadas virtuales (VPC) permitidas, donde cada punto de conexión de VPC se especifica como referencia, por ejemplo, una [referencia dinámica](#) o la [función intrínseca Ref](#). Para obtener un ejemplo del uso de esta propiedad, consulta la sección Ejemplos al final de esta página.

Tipo: lista

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

`IpRangeBlacklist`

Las direcciones IP o los rangos de direcciones que se van a bloquear. Para obtener un ejemplo del uso de esta propiedad, consulta la sección Ejemplos al final de esta página.

Tipo: lista

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

IpRangeWhitelist

Las direcciones IP o los rangos de direcciones que se van a permitir.

Tipo: lista

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

SourceVpcBlacklist

La VPC de origen o los puntos de conexión de VPC que se van a bloquear. Los nombres de la VPC de origen deben empezar por "vpc-" y los nombres de los puntos de conexión de VPC de origen deben empezar por "vpce-". Para obtener un ejemplo del uso de esta propiedad, consulta la sección Ejemplos al final de esta página.

Tipo: lista

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

SourceVpcWhitelist

La VPC de origen o los puntos de conexión de VPC que se van a permitir. Los nombres de la VPC de origen deben empezar por "vpc-" y los nombres de los puntos de conexión de VPC de origen deben empezar por "vpce-".

Tipo: lista

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

Ejemplos

Ejemplo de política de recursos

El siguiente ejemplo bloquea dos direcciones IP y una VPC de origen y permite una AWS cuenta.

YAML

```
Auth:
  ResourcePolicy:
    CustomStatements: [{
      "Effect": "Allow",
      "Principal": "*",
      "Action": "execute-api:Invoke",
      "Resource": "execute-api:/Prod/GET/pets",
      "Condition": {
        "IpAddress": {
          "aws:SourceIp": "1.2.3.4"
        }
      }
    }]

  IpRangeBlacklist:
    - "10.20.30.40"
    - "1.2.3.4"
  SourceVpcBlacklist:
    - "vpce-1a2b3c4d"
  AwsAccountWhitelist:
    - "111122223333"
  IntrinsicVpcBlacklist:
    - "{{resolve:ssm:SomeVPCReference:1}}"
    - !Ref MyVPC
  IntrinsicVpceWhitelist:
    - "{{resolve:ssm:SomeVPCEReference:1}}"
    - !Ref MyVPCE
```

RequestModel

Configura un modelo de solicitud para una API+ruta+método específicos.

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
Model: String
Required: Boolean
ValidateBody: Boolean
```

[ValidateParameters](#): *Boolean*

Propiedades

Model

Nombre de un modelo definido en la propiedad Modelos del [AWS::Serverless::API](#).

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

Required

Añade una `required` propiedad en la sección de parámetros de la OpenApi definición para el punto final de la API en cuestión.

Tipo: Booleano

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

ValidateBody

Especifica si API Gateway utiliza el `Model` para validar el cuerpo de la solicitud. Para obtener más información, consulta [Habilitar la validación de solicitudes en API Gateway](#) en la Guía para desarrolladores de API Gateway.

Tipo: Booleano

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

ValidateParameters

Especifica si API Gateway utiliza el `Model` para validar los parámetros de la ruta de solicitud, las cadenas de consulta y los encabezados. Para obtener más información, consulta [Habilitar la validación de solicitudes en API Gateway](#) en la Guía para desarrolladores de API Gateway.

Tipo: Booleano

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

Ejemplos

Modelo de solicitud

Ejemplo de modelo de solicitud

YAML

```
RequestModel:
  Model: User
  Required: true
  ValidateBody: true
  ValidateParameters: true
```

RequestParameter

Configura un parámetro de solicitud para una API+ruta+método específicos.

Se debe especificar una propiedad tanto `Required` como `Caching` para el parámetro de solicitud

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
Caching: Boolean
Required: Boolean
```

Propiedades

Caching

Añade una `cacheKeyParameters` sección a la OpenApi definición de API Gateway

Tipo: Booleano

Obligatorio: condicional

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

Required

Este campo especifica si un parámetro es necesario.

Tipo: Booleano

Obligatorio: condicional

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

Ejemplos

Parámetro de solicitud

Ejemplo de configuración de los parámetros de solicitud

YAML

```
RequestParameters:
  - method.request.header.Authorization:
    Required: true
    Caching: true
```

CloudWatchEvent

El objeto que describe un tipo de fuente de evento CloudWatchEvent.

AWS Serverless Application Model (AWS SAM) genera un [AWS::Events::Rule](#) recurso cuando se establece este tipo de evento.

Nota importante: [EventBridgeRule](#) es el tipo de fuente de eventos preferido para usar, en lugar de CloudWatchEvent. EventBridgeRule y CloudWatchEvent usa el mismo servicio, API y CloudFormation recursos subyacentes. Sin embargo, solo AWS SAM agregará soporte para nuevas funciones para EventBridgeRule.

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
Enabled: Boolean  
EventBusName: String  
Input: String  
InputPath: String  
Pattern: EventPattern  
State: String
```

Propiedades

Enabled

Indica si la regla está habilitada.

Para deshabilitar la regla, defina esta propiedad en `false`.

Note

Especifique la propiedad `Enabled` o `State`, pero no ambas.

Tipo: Booleano

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es similar a la [State](#) propiedad de un `AWS::Events::Rule` recurso. Si esta propiedad se establece en, `true` entonces, se AWS SAM aprueba; de lo contrario `ENABLED`, se aprueba `DISABLED`.

EventBusName

El bus de eventos que se asociará a esta regla. Si omite esta propiedad, AWS SAM utiliza el bus de eventos predeterminado.

Tipo: cadena

Obligatorio: no

Predeterminado: bus de eventos predeterminado

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [EventBusName](#) propiedad de un `AWS::Events::Rule` recurso.

Input

Texto JSON válido transmitido al destino. Si utiliza esta propiedad, no se transmitirá nada del propio texto del evento al destino.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Input](#) propiedad de un `AWS::Events::Rule Target` recurso.

InputPath

Cuando no desee pasar todo el evento coincidente al destino, utilice la propiedad `InputPath` para describir qué parte del evento pasar.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [InputPath](#) propiedad de un `AWS::Events::Rule Target` recurso.

Pattern

Describe qué eventos se dirigen al destino especificado. Para obtener más información, consulte [Eventos y patrones de eventos EventBridge en](#) la Guía del EventBridge usuario de Amazon.

Tipo: [EventPattern](#)

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [EventPattern](#) propiedad de un `AWS::Events::Rule` recurso.

State

El estado de la regla.

Valores aceptados DISABLED | ENABLED:

 Note

Especifique la propiedad `Enabled` o `State`, pero no ambas.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [State](#) propiedad de un `AWS::Events::Rule` recurso.

Ejemplos

CloudWatchEvent

A continuación, se muestra un ejemplo de un tipo de fuente de evento `CloudWatchEvent`.

YAML

```
CWEvent:
  Type: CloudWatchEvent
  Properties:
    Enabled: false
    Input: '{"Key": "Value"}'
    Pattern:
      detail:
        state:
          - running
```

CloudWatchLogs

El objeto que describe un tipo de fuente de evento `CloudWatchLogs`.

Este evento genera un recurso [AWS::Logs::SubscriptionFilter](#), especifica un filtro de suscripción y lo asocia con el grupo de registros especificado.

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
FilterPattern: String  
LogGroupName: String
```

Propiedades

FilterPattern

Las expresiones de filtrado que restringen lo que se entrega al AWS recurso de destino. Para obtener más información acerca de la sintaxis del patrón de filtro, consulte [Filter and Pattern Syntax](#).

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [FilterPattern](#) propiedad de un `AWS::Logs::SubscriptionFilter` recurso.

LogGroupName

El grupo de registros que se asociará al filtro de suscripción. Todos los eventos de registro que se cargan en este grupo de registros se filtran y se envían al AWS recurso especificado si el patrón de filtrado coincide con los eventos del registro.

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [LogGroupName](#) propiedad de un `AWS::Logs::SubscriptionFilter` recurso.

Ejemplos

Filtro de suscripción de Cloudwatchlogs

Ejemplo de filtro de suscripción de Cloudwatchlogs

YAML

```
CWLog:
```

```
Type: CloudWatchLogs
Properties:
  LogGroupName:
    Ref: CloudWatchLambdaLogsGroup
  FilterPattern: My pattern
```

Cognito

El objeto que describe un tipo de fuente de evento Cognito.

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
Trigger: List
UserPool: String
```

Propiedades

Trigger

La información de configuración del desencadenador de Lambda para el nuevo grupo de usuarios.

Tipo: lista

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [LambdaConfig](#) propiedad de un `AWS::Cognito::UserPool` recurso.

UserPool

Referencia a lo UserPool definido en la misma plantilla

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

Ejemplos

Cognito Event

Cognito Event Example

YAML

```
CognitoUserPoolPreSignup:
  Type: Cognito
  Properties:
    UserPool:
      Ref: MyCognitoUserPool
    Trigger: PreSignUp
```

DocumentDB

El objeto que describe un tipo de fuente de evento DocumentDB. Para obtener más información, consulte [Uso AWS Lambda con Amazon DocumentDB](#) en la Guía para AWS Lambda desarrolladores.

Sintaxis

Para declarar esta entidad en la AWS SAM plantilla, utilice la siguiente sintaxis.

YAML

```
BatchSize: Integer
Cluster: String
CollectionName: String
DatabaseName: String
Enabled: Boolean
FilterCriteria: FilterCriteria
FullDocument: String
KmsKeyArn: String
MaximumBatchingWindowInSeconds: Integer
SecretsManagerKmsKeyId: String
SourceAccessConfigurations: List
```

StartingPosition: *String*
StartingPositionTimestamp: *Double*

Propiedades

BatchSize

El número máximo de elementos que se recuperan en un único lote.

Tipo: entero

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [BatchSize](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

Cluster

El nombre de recurso de Amazon (ARN) del clúster de Amazon DocumentDB.

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [EventSourceArn](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

CollectionName

El nombre de la colección a consumir dentro de la base de datos. Si no especifica ninguna colección, Lambda consume todas las colecciones.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [CollectionName](#) propiedad de un tipo de `AWS::Lambda::EventSourceMapping` `DocumentDBEventSourceConfig` datos.

DatabaseName

El nombre de la base de datos a consumir dentro del clúster de Amazon DocumentDB.

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [DatabaseName](#) propiedad de un tipo de `AWS::Lambda::EventSourceMapping` `DocumentDBEventSourceConfig` datos.

Enabled

Si es `true`, la asignación de orígenes de eventos está activa. Establézcalo en `false` para poner en pausa el sondeo y la invocación.

Tipo: Booleano

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Enabled](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

FilterCriteria

Un objeto que define los criterios que determinan si Lambda debe procesar un evento o no. Para obtener más información, consulta [Filtrado de eventos de Lambda](#) en la Guía para desarrolladores de AWS Lambda .

Tipo: [FilterCriteria](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [FilterCriteria](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

FullDocument

Determina lo que Amazon DocumentDB envía al flujo de eventos durante las operaciones de actualización del documento. Si se establece en `UpdateLookup`, Amazon DocumentDB envía un delta que describe los cambios, junto con una copia del documento completo. De lo contrario, Amazon DocumentDB envía solo un documento parcial que contiene los cambios.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [FullDocument](#) propiedad de un tipo de `AWS::Lambda::EventSourceMapping` `DocumentDBEventSourceConfig` datos.

KmsKeyArn

El nombre de recurso de Amazon (ARN) de la clave de que se utilizará para cifrar la información de este evento.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [KmsKeyArn](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

MaximumBatchingWindowInSeconds

La cantidad de tiempo máxima para recopilar registros antes de invocar la función, en segundos.

Tipo: Entero

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [MaximumBatchingWindowInSeconds](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

SecretsManagerKmsKeyId

El identificador de clave AWS Key Management Service (AWS KMS) de una clave gestionada por el cliente de AWS Secrets Manager. Se requiere cuando se utiliza una clave gestionada por el cliente de Secrets Manager con un rol de ejecución de Lambda que no incluye el permiso `kms:Decrypt`.

El valor de esta propiedad es un UUID. Por ejemplo: `1abc23d4-567f-8ab9-cde0-1fab234c5d67`.

Tipo: cadena

Obligatorio: condicional

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

SourceAccessConfigurations

Matriz del protocolo de autenticación o del host virtual. Especifique esto mediante el tipo [SourceAccessConfigurations](#) de datos.

Para el tipo de origen del evento DocumentDB, el único tipo de configuración válido es BASIC_AUTH

- BASIC_AUTH: el secreto de Secrets Manager que almacena las credenciales del agente. Para este tipo, la credencial debe tener el siguiente formato: {"username": "your-username", "password": "your-password"}. Solo se permite un objeto de tipo BASIC_AUTH.

Tipo: lista

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [SourceAccessConfigurations](#) propiedad de un AWS::Lambda::EventSourceMapping recurso.

StartingPosition

La posición en el flujo donde comienza la lectura.

- AT_TIMESTAMP: especifique el tiempo a partir del cual comenzar la lectura de registros.
- LATEST: registros nuevos de solo lectura.
- TRIM_HORIZON: procese todos los registros disponibles.

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [StartingPosition](#) propiedad de un AWS::Lambda::EventSourceMapping recurso.

StartingPositionTimestamp

El tiempo a partir del cual comenzar la lectura, en segundos de tiempo Unix. Defina StartingPositionTimestamp cuando StartingPosition se especifica como AT_TIMESTAMP.

Tipo: Doble

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [StartingPositionTimestamp](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

Ejemplos

Origen de eventos de Amazon DocumentDB

```
AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31
...
Resources:
  MyFunction:
    Type: AWS::Serverless::Function
    Properties:
      ...
      Events:
        MyDDBEvent:
          Type: DocumentDB
          Properties:
            Cluster: "arn:aws:rds:us-west-2:123456789012:cluster:docdb-2023-01-01"
            BatchSize: 10
            MaximumBatchingWindowInSeconds: 5
            DatabaseName: "db1"
            CollectionName: "collection1"
            FullDocument: "UpdateLookup"
            SourceAccessConfigurations:
              - Type: BASIC_AUTH
                URI: "arn:aws:secretsmanager:us-west-2:123456789012:secret:doc-db"
```

DynamoDB

El objeto que describe un tipo de fuente de evento DynamoDB. Para obtener más información, consulte [Uso AWS Lambda con Amazon DynamoDB](#) en AWS Lambda la Guía para desarrolladores.

AWS SAM genera un [AWS::Lambda::EventSourceMapping](#) recurso cuando se establece este tipo de evento.

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
BatchSize: Integer  
BisectBatchOnFunctionError: Boolean  
DestinationConfig: DestinationConfig  
Enabled: Boolean  
FilterCriteria: FilterCriteria  
FunctionResponseTypes: List  
KmsKeyArn: String  
MaximumBatchingWindowInSeconds: Integer  
MaximumRecordAgeInSeconds: Integer  
MaximumRetryAttempts: Integer  
MetricsConfig: MetricsConfig  
ParallelizationFactor: Integer  
StartingPosition: String  
StartingPositionTimestamp: Double  
Stream: String  
TumblingWindowInSeconds: Integer
```

Propiedades

BatchSize

El número máximo de elementos que se recuperan en un único lote.

Tipo: entero

Obligatorio: no

Valor predeterminado: 100

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [BatchSize](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

Mínimo: 1

Máximo: 1000

BisectBatchOnFunctionError

Si la función devuelve un error, divida el lote en dos y vuelva a intentarlo.

Tipo: Booleano

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [BisectBatchOnFunctionError](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

DestinationConfig

Una cola de Amazon Simple Queue Service (Amazon SQS) o un destino de tema de Amazon Simple Notification Service (Amazon SNS) para registros descartados.

Tipo: [DestinationConfig](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [DestinationConfig](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

Enabled

Deshabilita el mapeo de origen de eventos para pausar el sondeo y la invocación.

Tipo: Booleano

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Enabled](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

FilterCriteria

Un objeto que define los criterios que determinan si Lambda debe procesar un evento o no. Para obtener más información, consulta [Filtrado de eventos de AWS Lambda](#) en la Guía para desarrolladores de AWS Lambda .

Tipo: [FilterCriteria](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [FilterCriteria](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

FunctionResponseTypes

Una lista de enumeraciones de tipos de respuesta actuales aplicadas a la asignación de origen de eventos. Para obtener más información, consulta [Informes de errores de artículos en lotes](#) en la Guía para desarrolladores de AWS Lambda .

Valores válidos: `ReportBatchItemFailures`

Tipo: lista

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [FunctionResponseTypes](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

`KmsKeyArn`

El nombre de recurso de Amazon (ARN) de la clave de que se utilizará para cifrar la información de este evento.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [KmsKeyArn](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

`MaximumBatchingWindowInSeconds`

La cantidad de tiempo máxima para recopilar registros antes de invocar la función, en segundos.

Tipo: Entero

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [MaximumBatchingWindowInSeconds](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

`MaximumRecordAgeInSeconds`

La antigüedad máxima de un registro que Lambda envía a una función para su procesamiento.

Tipo: entero

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [MaximumRecordAgeInSeconds](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

MaximumRetryAttempts

El número máximo de veces que se debe volver a intentar cuando la función devuelve un error.

Tipo: entero

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [MaximumRetryAttempts](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

MetricsConfig

Una configuración opcional para obtener métricas mejoradas para las asignaciones de orígenes de eventos que capturan cada etapa del procesamiento. Para ver un ejemplo, consulta [MetricsConfig evento](#).

Tipo: [MetricsConfig](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [MetricsConfig](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

ParallelizationFactor

El número de lotes de cada partición que se procesarán simultáneamente.

Tipo: entero

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [ParallelizationFactor](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

StartingPosition

La posición en el flujo donde comienza la lectura.

- `AT_TIMESTAMP`: especifique el tiempo a partir del cual comenzar la lectura de registros.
- `LATEST`: registros nuevos de solo lectura.
- `TRIM_HORIZON`: procese todos los registros disponibles.

Valores válidos: `AT_TIMESTAMP` | `LATEST` | `TRIM_HORIZON`

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [StartingPosition](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

StartingPositionTimestamp

El tiempo a partir del cual comenzar la lectura, en segundos de tiempo Unix. Defina `StartingPositionTimestamp` cuando `StartingPosition` se especifica como `AT_TIMESTAMP`.

Tipo: Doble

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [StartingPositionTimestamp](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

Stream

El nombre de recurso de Amazon (ARN) del flujo de DynamoDB.

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [EventSourceArn](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

TumblingWindowInSeconds

La duración, en segundos, de una ventana de procesamiento. El intervalo válido es de 1 a 900 (15 minutos).

Para más información, consulta [Ventanas móviles](#) en la Guía para desarrolladores de AWS Lambda .

Tipo: entero

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [TumblingWindowInSeconds](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

Ejemplos

MetricsConfig evento

El siguiente ejemplo muestra un recurso que usa la propiedad `MetricsConfig` para capturar cada etapa del procesamiento para sus asignaciones de orígenes de eventos.

```
Resources:
  FilteredEventsFunction:
    Type: AWS::Serverless::Function
    Properties:
      CodeUri: s3://sam-demo-bucket/metricsConfig.zip
      Handler: index.handler
      Runtime: nodejs16.x
      Events:
        KinesisStream:
          Type: Kinesis
          Properties:
            Stream: !GetAtt KinesisStream.Arn
            StartingPosition: LATEST
            MetricsConfig:
              Metrics:
                - EventCount
```

Origen de eventos de DynamoDB para la tabla de DynamoDB existente

Origen de eventos de DynamoDB para una tabla de DynamoDB que ya existe en una cuenta. AWS

YAML

```
Events:
  DDBEvent:
    Type: DynamoDB
    Properties:
      Stream: arn:aws:dynamodb:us-east-1:123456789012:table/TestTable/
stream/2016-08-11T21:21:33.291
      StartingPosition: TRIM_HORIZON
      BatchSize: 10
```



```
Enabled: false
```

Evento de DynamoDB para la tabla de DynamoDB declarado en la plantilla

Evento de DynamoDB para una tabla de DynamoDB declarada en el mismo archivo de plantilla.

YAML

```
Events:
  DDBEvent:
    Type: DynamoDB
    Properties:
      Stream:
        !GetAtt MyDynamoDBTable.StreamArn # This must be the name of a DynamoDB table
        declared in the same template file
      StartingPosition: TRIM_HORIZON
      BatchSize: 10
      Enabled: false
```

EventBridgeRule

El objeto que describe un tipo de fuente de EventBridgeRule eventos, que establece tu función sin servidor como el objetivo de una EventBridge regla de Amazon. Para obtener más información, consulta [¿Qué es Amazon EventBridge?](#) en la Guía del EventBridge usuario de Amazon.

AWS SAM genera un [AWS::Events::Rule](#) recurso cuando se establece este tipo de evento. AWS SAM también crea un `AWS::Lambda::Permission` recurso, que es necesario para EventBridgeRule poder llamar a Lambda.

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
DeadLetterConfig: DeadLetterConfig
EventBusName: String
Input: String
InputPath: String
```

`InputTransformer`: [InputTransformer](#)
`Pattern`: [EventPattern](#)
`RetryPolicy`: [RetryPolicy](#)
`RuleName`: *String*
`State`: *String*
`Target`: [Target](#)

Propiedades

DeadLetterConfig

Configure la cola del Amazon Simple Queue Service (Amazon SQS), desde la que EventBridge se envían los eventos tras una invocación de destino fallida. La invocación puede fallar, por ejemplo, cuando se envía un evento a una función de eventos de Lambda que no existe o EventBridge cuando no hay permisos suficientes para invocar la función de Lambda. Para obtener más información, consulta la [política de reintentos de eventos y el uso de colas con letra muerta en la Guía del usuario](#) de Amazon. EventBridge

Note

El tipo de recurso [AWS::Serverless::Función](#) tiene un tipo de datos similar, `DeadLetterQueue`, que gestiona los errores que se producen tras la invocación correcta de la función de Lambda de destino. Entre los ejemplos de estos tipos de errores se incluyen la limitación de Lambda o los errores devueltos por la función de destino de Lambda. Para obtener más información sobre la *DeadLetterQueue* propiedad de la función, consulte las [Dead-letter colas](#) en la Guía para desarrolladores.AWS Lambda

Tipo: [DeadLetterConfig](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es similar a la [DeadLetterConfig](#) propiedad del tipo de `AWS::Events::Rule` Target datos. La AWS SAM versión de esta propiedad incluye subpropiedades adicionales, por si AWS SAM quiere crear automáticamente una cola de texto sin procesar.

EventBusName

El bus de eventos que se asociará a esta regla. Si omite esta propiedad, AWS SAM utiliza el bus de eventos predeterminado.

Tipo: cadena

Obligatorio: no

Predeterminado: bus de eventos predeterminado

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [EventBusName](#) propiedad de un `AWS::Events::Rule` recurso.

Input

Texto JSON válido transmitido al destino. Si utiliza esta propiedad, no se transmitirá nada del propio texto del evento al destino.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Input](#) propiedad de un `AWS::Events::Rule` Target recurso.

InputPath

Cuando no desee pasar todo el evento coincidente al destino, utilice la propiedad `InputPath` para describir qué parte del evento pasar.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [InputPath](#) propiedad de un `AWS::Events::Rule` Target recurso.

InputTransformer

Configuración que permite proporcionar una entrada personalizada a un destino en función de determinados datos de eventos. Puedes extraer uno o más pares clave-valor del evento y, a continuación, utilizar dichos datos para enviar una entrada personalizada al destino. Para obtener más información, consulta la [transformación EventBridge de entradas de Amazon](#) en la Guía del EventBridge usuario de Amazon.

Tipo: [InputTransformer](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [InputTransformer](#) propiedad de un tipo de `AWS::Events::Rule` Target datos.

Pattern

Describe qué eventos se dirigen al destino especificado. Para obtener más información, consulta [EventBridge los eventos y patrones de EventBridge eventos de Amazon](#) en la Guía del EventBridge usuario de Amazon.

Tipo: [EventPattern](#)

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [EventPattern](#) propiedad de un `AWS::Events::Rule` recurso.

RetryPolicy

Objeto de la `RetryPolicy` que incluye información acerca de la configuración de la política de reintentos. Para obtener más información, consulta la [política de reintentos de eventos y el uso de colas con letra muerta en la Guía del usuario](#) de Amazon. EventBridge

Tipo: [RetryPolicy](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [RetryPolicy](#) propiedad del tipo de datos. `AWS::Events::Rule` Target

RuleName

El nombre de la regla.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Name](#) propiedad de un `AWS::Events::Rule` recurso.

State

El estado de la regla.

Valores aceptados: `DISABLED` | `ENABLED` | `ENABLED_WITH_ALL_CLOUDTRAIL_MANAGEMENT_EVENTS`

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [State](#) propiedad de un `AWS::Events::Rule` recurso.

Target

El AWS recurso que se EventBridge invoca cuando se activa una regla. Puedes utilizar esta propiedad para especificar el ID lógico del objetivo. Si no se especifica esta propiedad, AWS SAM genera el ID lógico del objetivo.

Tipo: [destino](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es similar a la [Targets](#) propiedad de un `AWS::Events::Rule` recurso. Amazon EC2 RebootInstances API calles un ejemplo de una propiedad de destino. La versión AWS SAM de esta propiedad solo permite especificar el identificador lógico de un único destino.

Ejemplos

EventBridgeRule

A continuación, se muestra un ejemplo de un tipo de fuente de evento EventBridgeRule.

YAML

```
EBRule:
  Type: EventBridgeRule
  Properties:
    Input: '{"Key": "Value"}'
    Pattern:
      detail:
        state:
          - terminated
    RetryPolicy:
      MaximumRetryAttempts: 5
      MaximumEventAgeInSeconds: 900
    DeadLetterConfig:
      Type: SQS
```

```
QueueLogicalId: EBRuleDLQ
Target:
  Id: MyTarget
```

DeadLetterConfig

El objeto utilizado para especificar la cola del Amazon Simple Queue Service (Amazon SQS) desde la que EventBridge se envían los eventos tras una invocación de destino fallida. La invocación puede fallar, por ejemplo, cuando se envía un evento a una función de eventos de Lambda que no existe o si no hay permisos suficientes para invocar la función de Lambda. Para obtener más información, consulta la [política de reintentos de eventos y el uso de colas con letra muerta en la Guía del usuario](#) de Amazon. EventBridge

Note

El tipo de recurso [AWS::Serverless::Función](#) tiene un tipo de datos similar, `DeadLetterQueue`, que gestiona los errores que se producen tras la invocación correcta de la función de Lambda de destino. Entre los ejemplos de este tipo de errores se incluyen la limitación de Lambda o los errores devueltos por la función de destino de Lambda. Para obtener más información sobre la `DeadLetterQueue` propiedad de la función, consulte las [Dead-letter colas](#) en la Guía para desarrolladores.AWS Lambda

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
Arn: String
QueueLogicalId: String
Type: String
```

Propiedades

Arn

El nombre de recurso de Amazon (ARN) de la cola SQS de Amazon especificada como destino de la cola de mensajes fallidos.

 Note

Especifique la propiedad `Type` o la propiedad `Arn`, pero no ambas.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Arn](#) propiedad del tipo de `AWS::Events::Rule DeadLetterConfig` datos.

QueueLogicalId

`Type` Se especifica el nombre personalizado de la cola de cartas muertas que la AWS SAM crea.

 Note

Si no se especifica la propiedad `Type`, entonces esta propiedad se pasa por alto.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

Type

Tipo de cola. Cuando se establece esta propiedad, crea AWS SAM automáticamente una cola sin efecto y adjunta la [política basada en los recursos](#) necesaria para conceder permiso al recurso de regla para enviar eventos a la cola.

 Note

Especifica la propiedad `Type` o la propiedad `Arn`, pero no ambas.

Valores válidos: `SQS`

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva y no tiene un equivalente. AWS SAM CloudFormation

Ejemplos

DeadLetterConfig

DeadLetterConfig

YAML

```
DeadLetterConfig:  
  Type: SQS  
  QueueLogicalId: MyDLQ
```

Target

Configura el AWS recurso que se EventBridge invoca cuando se activa una regla.

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
Id: String
```

Propiedades

Id

El ID lógico del destino.

El valor de Id puede contener caracteres alfanuméricos, puntos (.), guiones (-) y guiones bajos (_).

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Id](#) propiedad del tipo de `AWS::Events::Rule` Target datos.

Ejemplos

Target

YAML

```
EBRule:
  Type: EventBridgeRule
  Properties:
    Target:
      Id: MyTarget
```

HttpApi

El objeto que describe la fuente de un evento con el tipo `HttpApi`.

Si existe una `OpenApi` definición para la ruta y el método especificados en la API, SAM le agregará la sección de integración y seguridad de Lambda (si corresponde).

Si en la API no existe una `OpenApi` definición para la ruta y el método especificados, SAM creará esta definición automáticamente.

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
ApiId: String
Auth: HttpApiFunctionAuth
Method: String
Path: String
PayloadFormatVersion: String
RouteSettings: RouteSettings
TimeoutInMillis: Integer
```

Propiedades

ApiId

Identificador de un recurso [AWS::Serverless::HttpApi](#) definido en esta plantilla.

Si no está definido, se crea un [AWS::Serverless::HttpApi](#) recurso predeterminado llamado `ServerlessHttpApi` mediante un OpenApi documento generado que contiene una unión de todas las rutas y métodos definidos por los eventos de Api definidos en esta plantilla que no especifican un `ApiId`.

Esto no puede hacer referencia a un recurso [AWS::Serverless::HttpApi](#) definido en otra plantilla.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

Auth

Configuración de autenticación para esta API, ruta y método específicos.

Útil para anular las `API DefaultAuthorizer` o establecer la configuración de autenticación en una ruta individual cuando no se especifica ningún `DefaultAuthorizer`.

Tipo: [HttpApiFunctionAuth](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

Method

Método HTTP para el que se invoca esta función.

Si no, `Path` y `Method` se especifican, SAM creará una ruta de API predeterminada que enruta cualquier solicitud que no se asigne a un punto final diferente a esta función de eventos de Lambda. Solo puede existir una de estas rutas predeterminadas por API.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

Path

Ruta de URI para la que se invoca esta función. Debe empezar con /.

Si no, Path y Method se especifican, SAM creará una ruta de API predeterminada que enruta cualquier solicitud que no se asigne a un punto de conexión diferente a esta función de Lambda. Solo puede existir una de estas rutas predeterminadas por API.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

PayloadFormatVersion

Especifica el formato de la carga enviada a una integración.

NOTA: PayloadFormatVersion requiere que SAM modifique la definición de OpenAPI, por lo que solo funciona con los elementos en línea OpenApi definidos en la propiedad. DefinitionBody

Tipo: cadena

Obligatorio: no

Predeterminado: 2.0.

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

RouteSettings

La configuración de ruta por ruta para esta API HTTP. Para obtener más información sobre la configuración de rutas, consulte [AWS::ApiGatewayV2: :Stage RouteSettings](#) en la Guía para desarrolladores de API Gateway.

Nota: Si RouteSettings se especifican tanto en la fuente de HttpApi recursos como en la de eventos, las AWS SAM fusiona teniendo prioridad las propiedades de la fuente de eventos.

Tipo: [RouteSettings](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [RouteSettings](#) propiedad de un `AWS::ApiGatewayV2::Stage` recurso.

TimeoutInMillis

Tiempo de espera personalizado entre 50 y 29 000 milisegundos.

NOTA: TimeoutInMillis requiere que SAM modifique la definición de OpenAPI, por lo que solo funciona con los elementos en línea OpenApi definidos en la propiedad. `DefinitionBody`

Tipo: entero

Obligatorio: no

Predeterminado: 5000

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

Ejemplos

HttpApi Evento predeterminado

HttpApi Evento que utiliza la ruta por defecto. Todas las rutas y métodos no mapeados de esta API se dirigirán a este punto de conexión.

YAML

```
Events:
  HttpApiEvent:
    Type: HttpApi
```

HttpApi

HttpApi Evento que usa una ruta y un método específicos.

YAML

```
Events:
  HttpApiEvent:
    Type: HttpApi
    Properties:
      Path: /
```

```
Method: GET
```

HttpApi Autorización

HttpApi Evento que utiliza un autorizador.

YAML

```
Events:
  HttpApiEvent:
    Type: HttpApi
    Properties:
      Path: /authenticated
      Method: GET
      Auth:
        Authorizer: OpenIdAuth
        AuthorizationScopes:
          - scope1
          - scope2
```

HttpApiFunctionAuth

Configura la autorización a nivel de evento.

Configura la autenticación para una API + Ruta + Método específica

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
AuthorizationScopes: List
Authorizer: String
```

Propiedades

AuthorizationScopes

Los ámbitos de autorización que se aplicarán a esta API, ruta y método.

Los ámbitos que se muestran aquí anularán los ámbitos aplicados por el `DefaultAuthorizer`, si existe alguno.

Tipo: lista

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

Authorizer

Authorizer para una función específica. Para utilizar la autorización de IAM, especifique `AWS_IAM` y especifique `true` para `EnableIamAuthorizer` en la sección `Globals` de tu plantilla.

Si ha especificado un autorizador global en la API y quiere hacer pública una función específica, anule configurando `Authorizer` en `NONE`.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

Ejemplos

Autenticación de funciones

Especificar la autorización a nivel de función

YAML

```
Auth:
  Authorizer: OpenIdAuth
  AuthorizationScopes:
    - scope1
    - scope2
```

Autorización de IAM

Especifica la autorización de IAM a nivel de evento. Para utilizar la autorización `AWS_IAM` a nivel de evento, también debe especificar `true` para `EnableIamAuthorizer` en la sección `Globals` de tu plantilla. Para obtener más información, consulta [La sección Globals del AWS SAM plantilla](#).

YAML

```
Globals:
  HttpApi:
    Auth:
      EnableIamAuthorizer: true

Resources:
  HttpApiFunctionWithIamAuth:
    Type: AWS::Serverless::Function
    Properties:
      Events:
        ApiEvent:
          Type: HttpApi
          Properties:
            Path: /iam-auth
            Method: GET
            Auth:
              Authorizer: AWS_IAM
      Handler: index.handler
      InlineCode: |
        def handler(event, context):
            return {'body': 'HttpApiFunctionWithIamAuth', 'statusCode': 200}
      Runtime: python3.9
```

IoTRule

El objeto que describe un tipo de fuente de evento de `IoTRule`.

Crea un [AWS::IoT::TopicRule](#) recurso para declarar una AWS IoT regla. Para obtener más información, consulta la [Documentación de CloudFormation](#).

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
AwsIotSqlVersion: String
Sql: String
```

Propiedades

AwsIotSqlVersion

Versión del motor de reglas SQL que debe utilizarse al evaluar la regla.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [AwsIotSqlVersion](#) propiedad de un `AWS::IoT::TopicRule` `TopicRulePayload` recurso.

Sql

El enunciado SQL usado para consultar el tema. Para obtener más información, consulta [SQL Reference \(Referencia de SQL de AWS IoT\)](#) en la Guía para desarrolladores de AWS IoT .

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Sql](#) propiedad de un `AWS::IoT::TopicRule` `TopicRulePayload` recurso.

Ejemplos

Regla IOT

Ejemplo de regla IOT

YAML

```
IoTRule:
  Type: IoTRule
  Properties:
    Sql: SELECT * FROM 'topic/test'
```

Kinesis

El objeto que describe un tipo de fuente de evento Kinesis. Para obtener más información, consulte [Uso AWS Lambda con Amazon Kinesis](#) en la Guía para AWS Lambda desarrolladores.

AWS SAM genera un [AWS::Lambda::EventSourceMapping](#) recurso cuando se establece este tipo de evento.

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
BatchSize: Integer
BisectBatchOnFunctionError: Boolean
DestinationConfig: DestinationConfig
Enabled: Boolean
FilterCriteria: FilterCriteria
FunctionResponseType: List
KmsKeyArn: String
MaximumBatchingWindowInSeconds: Integer
MaximumRecordAgeInSeconds: Integer
MaximumRetryAttempts: Integer
MetricsConfig: MetricsConfig
ParallelizationFactor: Integer
StartingPosition: String
StartingPositionTimestamp: Double
Stream: String
TumblingWindowInSeconds: Integer
```

Propiedades

BatchSize

El número máximo de elementos que se recuperan en un único lote.

Tipo: entero

Obligatorio: no

Valor predeterminado: 100

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [BatchSize](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

Mínimo: 1

Máximo: 10000

BisectBatchOnFunctionError

Si la función devuelve un error, divida el lote en dos y vuelva a intentarlo.

Tipo: Booleano

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [BisectBatchOnFunctionError](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

DestinationConfig

Una cola de Amazon Simple Queue Service (Amazon SQS) o un destino de tema de Amazon Simple Notification Service (Amazon SNS) para registros descartados.

Tipo: [DestinationConfig](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [DestinationConfig](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

Enabled

Deshabilita el mapeo de origen de eventos para pausar el sondeo y la invocación.

Tipo: Booleano

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Enabled](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

FilterCriteria

Un objeto que define los criterios que determinan si Lambda debe procesar un evento o no. Para obtener más información, consulta [Filtrado de eventos de AWS Lambda](#) en la Guía para desarrolladores de AWS Lambda .

Tipo: [FilterCriteria](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [FilterCriteria](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

FunctionResponseTypes

Una lista de enumeraciones de tipos de respuesta actuales aplicadas a la asignación de origen de eventos. Para obtener más información, consulta [Informes de errores de artículos en lotes](#) en la Guía para desarrolladores de AWS Lambda .

Valores válidos: `ReportBatchItemFailures`

Tipo: lista

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [FunctionResponseTypes](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

KmsKeyArn

El nombre de recurso de Amazon (ARN) de la clave de que se utilizará para cifrar la información de este evento.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [KmsKeyArn](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

MaximumBatchingWindowInSeconds

La cantidad de tiempo máxima para recopilar registros antes de invocar la función, en segundos.

Tipo: Entero

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [MaximumBatchingWindowInSeconds](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

MaximumRecordAgeInSeconds

La antigüedad máxima de un registro que Lambda envía a una función para su procesamiento.

Tipo: entero

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [MaximumRecordAgeInSeconds](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

MaximumRetryAttempts

El número máximo de veces que se debe volver a intentar cuando la función devuelve un error.

Tipo: entero

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [MaximumRetryAttempts](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

MetricsConfig

Una configuración opcional para obtener métricas mejoradas para las asignaciones de orígenes de eventos que capturan cada etapa del procesamiento. Para ver un ejemplo, consulta [MetricsConfig evento](#).

Tipo: [MetricsConfig](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [MetricsConfig](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

ParallelizationFactor

El número de lotes de cada partición que se procesarán simultáneamente.

Tipo: entero

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [ParallelizationFactor](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

StartingPosition

La posición en el flujo donde comienza la lectura.

- `AT_TIMESTAMP`: especifique el tiempo a partir del cual comenzar la lectura de registros.
- `LATEST`: registros nuevos de solo lectura.
- `TRIM_HORIZON`: procese todos los registros disponibles.

Valores válidos: `AT_TIMESTAMP` | `LATEST` | `TRIM_HORIZON`

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [StartingPosition](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

StartingPositionTimestamp

El tiempo a partir del cual comenzar la lectura, en segundos de tiempo Unix. Defina `StartingPositionTimestamp` cuando `StartingPosition` se especifica como `AT_TIMESTAMP`.

Tipo: Doble

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [StartingPositionTimestamp](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

Stream

El nombre de recurso de Amazon (ARN) del flujo de datos un consumidor de flujos.

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [EventSourceArn](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

TumblingWindowInSeconds

La duración, en segundos, de una ventana de procesamiento. El intervalo válido es de 1 a 900 (15 minutos).

Para más información, consulta [Ventanas móviles](#) en la Guía para desarrolladores de AWS Lambda .

Tipo: entero

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [TumblingWindowInSeconds](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

Ejemplos

MetricsConfig evento

El siguiente ejemplo muestra un recurso que usa la propiedad `MetricsConfig` para capturar cada etapa del procesamiento para sus asignaciones de orígenes de eventos.

```
Resources:
  FilteredEventsFunction:
    Type: AWS::Serverless::Function
    Properties:
      CodeUri: s3://sam-demo-bucket/metricsConfig.zip
      Handler: index.handler
      Runtime: nodejs16.x
      Events:
        KinesisStream:
          Type: Kinesis
          Properties:
            Stream: !GetAtt KinesisStream.Arn
            StartingPosition: LATEST
            MetricsConfig:
              Metrics:
                - EventCount
```

Origen de eventos de Kinesis

A continuación, se muestra un ejemplo de origen de evento de Kinesis.

YAML

```
Events:
```

```
KinesisEvent:
  Type: Kinesis
  Properties:
    Stream: arn:aws:kinesis:us-east-1:123456789012:stream/my-stream
    StartingPosition: TRIM_HORIZON
    BatchSize: 10
    Enabled: false
    FilterCriteria:
      Filters:
        - Pattern: '{"key": ["val1", "val2"]}'
```

MQ

El objeto que describe un tipo de fuente de evento de MQ. Para obtener más información, consulta [Uso de Lambda con Amazon MQ](#) en la Guía para desarrolladores de AWS Lambda .

AWS Serverless Application Model (AWS SAM) genera un [AWS::Lambda::EventSourceMapping](#) recurso cuando se establece este tipo de evento.

Note

Para tener una cola de Amazon MQ en una nube privada virtual (VPC) que se conecte a una función de eventos de Lambda en una red pública, la función de ejecución de la función debe incluir los siguientes permisos:

- `ec2:CreateNetworkInterface`
- `ec2>DeleteNetworkInterface`
- `ec2:DescribeNetworkInterfaces`
- `ec2:DescribeSecurityGroups`
- `ec2:DescribeSubnets`
- `ec2:DescribeVpcs`

Para obtener más información, consulta [Permisos del rol de ejecución](#) en la Guía para desarrolladores de AWS Lambda .

Sintaxis

Para declarar esta entidad en la AWS SAM plantilla, utilice la siguiente sintaxis.

YAML

```
BatchSize: Integer  
Broker: String  
DynamicPolicyName: Boolean  
Enabled: Boolean  
FilterCriteria: FilterCriteria  
KmsKeyArn: String  
MaximumBatchingWindowInSeconds: Integer  
Queues: List  
SecretsManagerKmsKeyId: String  
SourceAccessConfigurations: List
```

Propiedades

BatchSize

El número máximo de elementos que se recuperan en un único lote.

Tipo: entero

Obligatorio: no

Valor predeterminado: 100

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [BatchSize](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

Mínimo: 1

Máximo: 10000

Broker

El nombre de recurso de Amazon (ARN) del agente de Amazon MQ.

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [EventSourceArn](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

DynamicPolicyName

De forma predeterminada, el nombre de la política AWS Identity and Access Management (IAM) es `SamAutoGeneratedAMQPolicy` de compatibilidad con versiones anteriores. Especifica `true` para que utilice un nombre generado automáticamente para su política de IAM. Este nombre incluirá el ID lógico del origen del evento de Amazon MQ.

Note

Si utiliza más de un origen de eventos de Amazon MQ, especifique `true` para que no se dupliquen los nombres de las políticas de IAM.

Tipo: Booleano

Obligatorio: no

Valor predeterminado: `false`

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

Enabled

Si es `true`, la asignación de orígenes de eventos está activa. Establézcalo en `false` para poner en pausa el sondeo y la invocación.

Tipo: Booleano

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Enabled](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

FilterCriteria

Un objeto que define los criterios que determinan si Lambda debe procesar un evento o no. Para obtener más información, consulta [Filtrado de eventos de AWS Lambda](#) en la Guía para desarrolladores de AWS Lambda .

Tipo: [FilterCriteria](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [FilterCriteria](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

KmsKeyArn

El nombre de recurso de Amazon (ARN) de la clave de que se utilizará para cifrar la información de este evento.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [KmsKeyArn](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

MaximumBatchingWindowInSeconds

La cantidad de tiempo máxima para recopilar registros antes de invocar la función, en segundos.

Tipo: Entero

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [MaximumBatchingWindowInSeconds](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

Queues

Nombre de la cola de destino del agente de Amazon MQ que se va a consumir.

Tipo: lista

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Queues](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

SecretsManagerKmsKeyId

El identificador de clave AWS Key Management Service (AWS KMS) de una clave gestionada por el cliente desde AWS Secrets Manager. Se requiere cuando se utiliza una clave gestionada por el cliente de Secrets Manager con una función de ejecución de Lambda que no incluye el permiso de `kms:Decrypt`.

El valor de esta propiedad es un UUID. Por ejemplo: 1abc23d4-567f-8ab9-cde0-1fab234c5d67.

Tipo: cadena

Obligatorio: condicional

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

SourceAccessConfigurations

Una matriz del protocolo de autenticación para un host virtual. Especifique esto mediante el tipo [SourceAccessConfigurations](#) de datos.

Para el tipo de origen del evento de MQ, los únicos tipos de configuración válidos son BASIC_AUTH y VIRTUAL_HOST.

- **BASIC_AUTH**: el secreto de Secrets Manager que almacena las credenciales del agente. Para este tipo, la credencial debe tener el siguiente formato: {"username": "your-username", "password": "your-password"}. Solo se permite un objeto de tipo BASIC_AUTH.
- **VIRTUAL_HOST**: el nombre del host virtual en el agente de RabbitMQ. Lambda utilizará este host de RabbitMQ como el origen de eventos. Solo se permite un objeto de tipo VIRTUAL_HOST.

Tipo: lista

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [SourceAccessConfigurations](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

Ejemplos

Origen de eventos de Amazon MQ

A continuación, se muestra un ejemplo de un tipo de origen de evento MQ para un agente de Amazon MQ.

YAML

```
Events:
```

```
MQEvent:
  Type: MQ
  Properties:
    Broker: arn:aws:mq:us-east-2:123456789012:broker:MyBroker:b-1234a5b6-78cd-901e-2fgh-3i45j6k17819
    Queues: List of queues
    SourceAccessConfigurations:
      - Type: BASIC_AUTH
        URI: arn:aws:secretsmanager:us-east-1:01234567890:secret:MyBrokerSecretName
    BatchSize: 200
    Enabled: true
```

MSK

El objeto que describe un tipo de fuente de evento de MSK. Para obtener más información, consulte [Uso AWS Lambda con Amazon MSK](#) en la Guía para AWS Lambda desarrolladores.

AWS Serverless Application Model (AWS SAM) genera un [AWS::Lambda::EventSourceMapping](#) recurso cuando se establece este tipo de evento.

Para utilizar Schema Registry, debe definir los permisos de rol de IAM específicos para su función. Consulte la [Configuración completa con roles de IAM](#) para ver un ejemplo de la configuración necesaria.

Sintaxis

Para declarar esta entidad en la AWS SAM plantilla, utilice la siguiente sintaxis.

YAML

```
BatchSize: Integer
BisectBatchOnFunctionError: Boolean
ConsumerGroupId: String
DestinationConfig: DestinationConfig
Enabled: Boolean
FilterCriteria: FilterCriteria
FunctionResponseType: List
KmsKeyArn: String
MaximumBatchingWindowInSeconds: Integer
MaximumRecordAgeInSeconds: Integer
MaximumRetryAttempts: Integer
LoggingConfig: LoggingConfig
MetricsConfig: MetricsConfig
```

[ProvisionedPollerConfig](#): [ProvisionedPollerConfig](#)
[SchemaRegistryConfig](#): [SchemaRegistryConfig](#)
[SourceAccessConfigurations](#): [SourceAccessConfigurations](#)
[StartingPosition](#): *String*
[StartingPositionTimestamp](#): *Double*
[Stream](#): *String*
[Topics](#): *List*

Propiedades

BatchSize

El número máximo de registros en cada lote que Lambda extrae del flujo o la cola y envía a su función. Lambda pasa todos los registros del lote a la función en una sola llamada hasta el límite de carga para la invocación síncrona (6 MB).

Valor predeterminado: 100

Rango válido: valor mínimo de 1. Valor máximo de 10 000.

Tipo: entero

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [BatchSize](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

BisectBatchOnFunctionError

Si la función devuelve un error, divida el lote en dos y vuelva a intentarlo.

Tipo: Booleano

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [BisectBatchOnFunctionError](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

ConsumerGroupId

Cadena que configura cómo se leerán los eventos de los temas de Kafka.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [AmazonManagedKafkaConfiguration](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

DestinationConfig

Un objeto de configuración que especifica el destino de un evento después de que Lambda lo procese.

Utilice esta propiedad para especificar el destino de las invocaciones fallidas desde el origen de eventos de Amazon MSK.

Tipo: [DestinationConfig](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [DestinationConfig](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

Enabled

Deshabilita el mapeo de origen de eventos para pausar el sondeo y la invocación.

Tipo: Booleano

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Enabled](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

FilterCriteria

Un objeto que define los criterios que determinan si Lambda debe procesar un evento o no. Para obtener más información, consulta [Filtrado de eventos de AWS Lambda](#) en la Guía para desarrolladores de AWS Lambda .

Tipo: [FilterCriteria](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [FilterCriteria](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

FunctionResponseTypes

Una lista de enumeraciones de tipos de respuesta actuales aplicadas a la asignación de origen de eventos. Para obtener más información, consulta [Informes de errores de artículos en lotes](#) en la Guía para desarrolladores de AWS Lambda .

Valores válidos: ReportBatchItemFailures

Tipo: lista

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [FunctionResponseTypes](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

KmsKeyArn

El nombre de recurso de Amazon (ARN) de la clave de que se utilizará para cifrar la información de este evento.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [KmsKeyArn](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

MaximumBatchingWindowInSeconds

La cantidad de tiempo máxima para recopilar registros antes de invocar la función, en segundos.

Tipo: Entero

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [MaximumBatchingWindowInSeconds](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

MaximumRecordAgeInSeconds

La antigüedad máxima de un registro que Lambda envía a una función para su procesamiento.

Tipo: entero

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [MaximumRecordAgeInSeconds](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

MaximumRetryAttempts

El número máximo de veces que se debe volver a intentar cuando la función devuelve un error.

Tipo: entero

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [MaximumRetryAttempts](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

LoggingConfig

Objeto de configuración que especifica la configuración de registro para el mapeo de la fuente del evento.

Tipo: [LoggingConfig](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [LoggingConfig](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

MetricsConfig

Un objeto de configuración que especifica la configuración de las métricas para el mapeo de la fuente del evento.

Tipo: [MetricsConfig](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [MetricsConfig](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

ProvisionedPollerConfig

Configuración para aumentar la cantidad de sondeos utilizados para calcular las asignaciones de orígenes de eventos. Esta configuración permite un mínimo de 1 sondeador y un máximo de 2000 sondeadores. Para ver un ejemplo, consulte [ProvisionedPollerConfig ejemplo](#).

Tipo: [ProvisionedPollerConfig](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [ProvisionedPollerConfig](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

SchemaRegistryConfig

Configuración para usar un registro de esquemas con el origen de eventos de Kafka.

Note

Esta característica solicita que se configure `ProvisionedPollerConfig`.

Tipo: `SchemaRegistryConfig`

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [AmazonManagedKafkaEventSourceConfig](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

SourceAccessConfigurations

Una matriz del protocolo de autenticación, los componentes de VPC o el host virtual para proteger y definir su origen de eventos.

Valores válidos: `CLIENT_CERTIFICATE_TLS_AUTH`

Tipo: lista de [SourceAccessConfiguration](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad forma parte de la [AmazonManagedKafkaEventSourceConfig](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

StartingPosition

La posición en el flujo donde comienza la lectura.

- `AT_TIMESTAMP`: especifique el tiempo a partir del cual comenzar la lectura de registros.
- `LATEST`: registros nuevos de solo lectura.

- TRIM_HORIZON: procese todos los registros disponibles.

Valores válidos: AT_TIMESTAMP | LATEST | TRIM_HORIZON

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [StartingPosition](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

StartingPositionTimestamp

El tiempo a partir del cual comenzar la lectura, en segundos de tiempo Unix. Defina `StartingPositionTimestamp` cuando `StartingPosition` se especifica como `AT_TIMESTAMP`.

Tipo: Doble

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [StartingPositionTimestamp](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

Stream

El nombre de recurso de Amazon (ARN) del flujo de datos un consumidor de flujos.

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [EventSourceArn](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

Topics

El nombre del tema de Kafka.

Tipo: lista

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Topics](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

Ejemplos

Configuración completa con los roles de IAM

El siguiente ejemplo muestra una configuración completa que incluye la configuración del rol de IAM necesaria para usar Schema Registry:

```
Parameters:
  PreCreatedSubnetOne:
    Type: String
  PreCreatedSubnetTwo:
    Type: String
  MskClusterName4:
    Type: String

Resources:
  MyLambdaExecutionRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Version: '2012-10-17'
        Statement:
          - Action: [sts:AssumeRole]
            Effect: Allow
            Principal:
              Service: [lambda.amazonaws.com]
      Policies:
        - PolicyName: KafkaClusterPermissions
          PolicyDocument:
            Statement:
              - Action: [kafka:DescribeClusterV2, kafka:GetBootstrapBrokers]
                Effect: Allow
                Resource: 'arn:aws:kafka:us-east-1:123456789012:cluster/*'
        - PolicyName: KafkaAuthPolicy
          PolicyDocument:
            Statement:
              - Action: [secretsmanager:GetSecretValue, kms:Decrypt]
                Effect: "Allow"
                Resource: ['arn:aws:secretsmanager:us-west-2:123456789012:secret:kafkaSecret-*****',
                          'arn:aws:kms:us-west-2:123456789012:key/keyId']
        - PolicyName: ENIPolicy
          PolicyDocument:
            Statement:
```

```

    - Action: [ec2:CreateNetworkInterface,
              ec2:DescribeNetworkInterfaces, ec2:DescribeVpcs,
ec2:DeleteNetworkInterface,
              ec2:DescribeSubnets, ec2:DescribeSecurityGroups]
    Effect: Allow
    Resource: '*'
  - PolicyName: SchemaRegistryPolicy
    PolicyDocument:
      Statement:
        - Action: [glue:GetRegistry]
          Effect: Allow
          Resource: 'arn:aws:glue:{region}:{account-id}:registry/{registry-name}'
  - PolicyName: SchemaVersionsPolicy
    PolicyDocument:
      Statement:
        - Action: [glue:GetSchemaVersions]
          Effect: Allow
          Resource: '*'
  ManagedPolicyArns:
    - !Sub arn:${AWS::Partition}:iam::aws:policy/service-role/
AWSLambdaBasicExecutionRole
  Tags:
    - {Value: SAM, Key: lambda:createdBy}

MyMskCluster:
  Type: AWS::MSK::Cluster
  Properties:
    BrokerNodeGroupInfo:
      ClientSubnets:
        - Ref: PreCreatedSubnetOne
        - Ref: PreCreatedSubnetTwo
      InstanceType: kafka.t3.small
      StorageInfo:
        EBSSStorageInfo:
          VolumeSize: 1
    ClusterName:
      Ref: MskClusterName4
    KafkaVersion: 3.8.x
    NumberOfBrokerNodes: 2

MyMskStreamProcessor:
  Type: AWS::Serverless::Function
  Properties:
    Runtime: nodejs18.x

```

```

Handler: index.handler
CodeUri: ${codeuri}
Role:
  Fn::GetAtt: [MyLambdaExecutionRole, Arn]
Events:
  MyMskEvent:
    Type: MSK
    Properties:
      StartingPosition: LATEST
      Stream:
        Ref: MyMskCluster
      SourceAccessConfigurations:
        - Type: SASL_SCRAM_512_AUTH
          URI: !Sub arn:${AWS::Partition}:secretsmanager:us-
west-2:123456789012:secret:my-path/my-secret-name-1a2b3c
      Topics:
        - SchemaRegistryTestTopic
      ProvisionedPollerConfig:
        MinimumPollers: 1
      SchemaRegistryConfig:
        AccessConfigs:
          - Type: BASIC_AUTH
            URI: !Sub arn:${AWS::Partition}:secretsmanager:us-
west-2:123456789012:secret:my-path/my-secret-name-1a2b3c
        SchemaValidationConfigs:
          - Attribute: KEY
            EventRecordFormat: JSON
        SchemaRegistryURI: !Sub arn:${AWS::Partition}:glue:us-
west-2:123456789012:registry/myregistry

```

ProvisionedPollerConfig ejemplo

```

ProvisionedPollerConfig:
  MinimumPollers: 1
  MaximumPollers: 200

```

Ejemplo de Amazon MSK para un clúster existente

El siguiente es un ejemplo de un tipo de origen de eventos MSK para un clúster de Amazon MSK que ya existe en un Cuenta de AWS.

YAML

```
Events:
  MSKEvent:
    Type: MSK
    Properties:
      StartingPosition: LATEST
      Stream: arn:aws:kafka:us-east-1:012345678012:cluster/exampleClusterName/
abcdefab-1234-abcd-5678-cdef0123ab01-2
    Topics:
      - MyTopic
```

Ejemplo de Amazon MSK para un clúster declarado en la misma plantilla

El siguiente es un ejemplo de un tipo de origen de eventos MSK para un clúster de Amazon MSK que está declarado en el mismo archivo de plantilla.

YAML

```
Events:
  MSKEvent:
    Type: MSK
    Properties:
      StartingPosition: LATEST
      Stream:
        Ref: MyMskCluster    # This must be the name of an MSK cluster declared in the
same template file
    Topics:
      - MyTopic
```

Origen de eventos de MSK con Schema Registry

A continuación, se muestra un ejemplo de un tipo de origen de evento de MSK configurado con Schema Registry.

```
Events:
  MSKEvent:
    Type: MSK
    Properties:
      StartingPosition: LATEST
      Stream:
```

```

    Ref: MyMskCluster
  Topics:
    - SchemaRegistryTestTopic
  ProvisionedPollerConfig:
    MinimumPollers: 1
  SchemaRegistryConfig:
    SchemaRegistryURI: !Sub arn:${AWS::Partition}:glue:us-
west-2:123456789012:registry/myregistry
    EventRecordFormat: JSON
    SchemaValidationConfigs:
      - Attribute: KEY
      - Attribute: VALUE

```

Origen de eventos de MSK con Schema Registry confluentes

A continuación, se muestra un ejemplo de un tipo de origen de evento de MSK configurado con Schema Registry confluyente.

```

Events:
  MSKEvent:
    Type: MSK
    Properties:
      StartingPosition: LATEST
      Stream:
        Ref: MyMskCluster
      Topics:
        - SchemaRegistryTestTopic
      ProvisionedPollerConfig:
        MinimumPollers: 1
      SchemaRegistryConfig:
        SchemaRegistryURI: https://my-schema-registry.confluent.cloud
        AccessConfigs:
          - Type: BASIC_AUTH
            URI: !Sub arn:${AWS::Partition}:secretsmanager:us-
west-2:123456789012:secret:my-secret
        EventRecordFormat: JSON
        SchemaValidationConfigs:
          - Attribute: KEY
          - Attribute: VALUE

```

S3

El objeto que describe un tipo de fuente de evento de S3.

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
Bucket: String  
Events: String | List  
Filter: NotificationFilter
```

Propiedades

Bucket

Nombre del bucket de S3. Este bucket debe existir en la misma plantilla.

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad es similar a la [BucketName](#) propiedad de un `AWS::S3::Bucket` recurso. Este campo es obligatorio en SAM. Este campo solo acepta una referencia al bucket de S3 creado en esta plantilla

Events

El evento de bucket de Amazon S3 para el que se invoca la función de eventos de Lambda. Consulte los [tipos de eventos admitidos con Amazon S3](#) para obtener una lista de valores válidos.

Tipo: Cadenas | Lista

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [Event](#) propiedad del tipo de `AWS::S3::Bucket LambdaConfiguration` datos.

Filter

Son las reglas de filtrado que determinan qué objetos de Amazon S3 debe invocar la función de Lambda. Para obtener más información acerca del filtrado de nombres de clave de Amazon S3, consulta [Configuración de notificaciones de eventos de Amazon S3](#) en la Guía del desarrollador del usuario de Amazon Simple Storage Service.

Tipo: [NotificationFilter](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Filter](#) propiedad del tipo de `AWS::S3::Bucket` LambdaConfiguration datos.

Ejemplos

S3-Event

Ejemplo de un evento de S3.

YAML

```
Events:
  S3Event:
    Type: S3
    Properties:
      Bucket:
        Ref: ImagesBucket      # This must be the name of an S3 bucket declared in the
same template file
      Events: s3:ObjectCreated:*
      Filter:
        S3Key:
          Rules:
            - Name: prefix      # or "suffix"
              Value: value      # The value to search for in the S3 object key names
```

Schedule

El objeto que describe un tipo de fuente de Schedule eventos, que establece tu función sin servidor como el objetivo de una EventBridge regla de Amazon que se activa de forma programada. Para obtener más información, consulta [¿Qué es Amazon EventBridge?](#) en la Guía del EventBridge usuario de Amazon.

AWS Serverless Application Model (AWS SAM) genera un [AWS::Events::Rule](#) recurso cuando se establece este tipo de evento.

Note

EventBridge ahora ofrece una nueva capacidad de programación, [Amazon EventBridge Scheduler](#). Amazon EventBridge Scheduler es un programador sin servidor que te permite crear, ejecutar y gestionar tareas desde un servicio gestionado centralizado. EventBridge

Scheduleres altamente personalizable y ofrece una escalabilidad mejorada en comparación con las reglas EventBridge programadas, con un conjunto más amplio de operaciones de API objetivo y. Servicios de AWS

Le recomendamos que utilice EventBridge Scheduler para invocar objetivos según una programación. Para definir este tipo de fuente de eventos en sus AWS SAM plantillas, consulte [ScheduleV2](#).

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
DeadLetterConfig: DeadLetterConfig
Description: String
Enabled: Boolean
Input: String
Name: String
RetryPolicy: RetryPolicy
Schedule: String
State: String
```

Propiedades

DeadLetterConfig

Configure la cola del Amazon Simple Queue Service (Amazon SQS), desde la que EventBridge se envían los eventos tras una invocación de destino fallida. La invocación puede fallar, por ejemplo, cuando se envía un evento a una función de eventos de Lambda que no existe o EventBridge cuando no hay permisos suficientes para invocar la función de Lambda. Para obtener más información, consulta la [política de reintentos de eventos y el uso de colas con letra muerta en la Guía del usuario](#) de Amazon. EventBridge

Note

El tipo de recurso [AWS::Serverless::Función](#) tiene un tipo de datos similar, `DeadLetterQueue`, que gestiona los errores que se producen tras la invocación correcta de la función de Lambda de destino. Entre los ejemplos de estos tipos de errores se

incluyen la limitación de Lambda o los errores devueltos por la función de destino de Lambda. Para obtener más información sobre la *DeadLetterQueue* propiedad de la función, consulte las [Dead-letter colas](#) en la Guía para desarrolladores.AWS Lambda

Tipo: [DeadLetterConfig](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es similar a la [DeadLetterConfig](#) propiedad del tipo de `AWS::Events::Rule` Target datos. La AWS SAM versión de esta propiedad incluye subpropiedades adicionales, por si AWS SAM quiere crear automáticamente una cola de texto sin procesar.

Description

Una descripción del conjunto de reglas.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [Description](#) propiedad de un recurso. `AWS::Events::Rule`

Enabled

Indica si la regla está habilitada.

Para deshabilitar la regla, defina esta propiedad en `false`.

Note

Especifique la propiedad `Enabled` o `State`, pero no ambas.

Tipo: Booleano

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es similar a la [State](#) propiedad de un `AWS::Events::Rule` recurso. Si esta propiedad se establece en, `true` entonces, se AWS SAM aprueba; de lo contrario `ENABLED`, se aprueba `DISABLED`.

Input

Texto JSON válido transmitido al destino. Si utiliza esta propiedad, no se transmitirá nada del propio texto del evento al destino.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Input](#) propiedad de un `AWS::Events::Rule` Target recurso.

Name

El nombre de la regla. Si no especificas un nombre, CloudFormation genera un identificador físico único y lo utiliza como nombre de la regla.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [Name](#) propiedad de un `AWS::Events::Rule` recurso.

RetryPolicy

Objeto de la `RetryPolicy` que incluye información acerca de la configuración de la política de reintentos. Para obtener más información, consulta la [política de reintentos de eventos y el uso de colas con letra muerta en la Guía del usuario](#) de Amazon. EventBridge

Tipo: [RetryPolicy](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [RetryPolicy](#) propiedad del tipo de datos. `AWS::Events::Rule` Target

Schedule

La expresión de programación que determina cuándo y con qué frecuencia se ejecuta la regla. Para obtener más información, consulta [Expresiones de programación para reglas](#).

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [ScheduleExpression](#) propiedad de un `AWS::Events::Rule` recurso.

State

El estado de la regla.

Valores aceptados DISABLED | ENABLED:

Note

Especifique la propiedad Enabled o State, pero no ambas.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [State](#) propiedad de un `AWS::Events::Rule` recurso.

Ejemplos

CloudWatch Programar evento

CloudWatch Ejemplo de programar un evento

YAML

```
CWSchedule:
  Type: Schedule
  Properties:
    Schedule: 'rate(1 minute)'
    Name: TestSchedule
    Description: test schedule
    Enabled: false
```

DeadLetterConfig

El objeto utilizado para especificar la cola del Amazon Simple Queue Service (Amazon SQS) desde la que EventBridge se envían los eventos tras una invocación de destino fallida. La invocación puede

fallar, por ejemplo, cuando se envía un evento a una función de eventos de Lambda que no existe o si no hay permisos suficientes para invocar la función de Lambda. Para obtener más información, consulta la [política de reintentos de eventos y el uso de colas con letra muerta en la Guía del usuario de Amazon EventBridge](#).

Note

El tipo de recurso `AWS::Serverless::Función` tiene un tipo de datos similar, `DeadLetterQueue`, que gestiona los errores que se producen tras la invocación correcta de la función de Lambda de destino. Entre los ejemplos de este tipo de errores se incluyen la limitación de Lambda o los errores devueltos por la función de destino de Lambda. Para obtener más información sobre la `DeadLetterQueue` propiedad de la función, consulte las [Dead-letter colas](#) en la Guía para desarrolladores AWS Lambda.

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
Arn: String
QueueLogicalId: String
Type: String
```

Propiedades

Arn

El nombre de recurso de Amazon (ARN) de la cola SQS de Amazon especificada como destino de la cola de mensajes fallidos.

Note

Especifique la propiedad `Type` o la propiedad `Arn`, pero no ambas.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Arn](#) propiedad del tipo de `AWS::Events::Rule` DeadLetterConfig datos.

QueueLogicalId

TypeSe especifica el nombre personalizado de la cola de cartas muertas que la AWS SAM crea.

 Note

Si no se especifica la propiedad Type, entonces esta propiedad se pasa por alto.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

Type

Tipo de cola. Cuando se establece esta propiedad, crea AWS SAM automáticamente una cola sin efecto y adjunta la [política basada en los recursos](#) necesaria para conceder permiso al recurso de regla para enviar eventos a la cola.

 Note

Especifica la propiedad Type o la propiedad Arn, pero no ambas.

Valores válidos: SQS

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva y no tiene un equivalente. AWS SAM CloudFormation

Ejemplos

DeadLetterConfig

DeadLetterConfig

YAML

```
DeadLetterConfig:
  Type: SQS
  QueueLogicalId: MyDLQ
```

ScheduleV2

El objeto que describe un tipo de fuente de `ScheduleV2` eventos, que establece su función sin servidor como el destino de un evento de Amazon EventBridge Scheduler que se desencadena de forma programada. Para obtener más información, consulte [¿Qué es Amazon EventBridge Scheduler?](#) en la Guía del usuario de EventBridge Scheduler.

AWS Serverless Application Model (AWS SAM) genera un [AWS::Scheduler::Schedule](#) recurso cuando se establece este tipo de evento.

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
DeadLetterConfig: DeadLetterConfig
Description: String
EndDate: String
FlexibleTimeWindow: FlexibleTimeWindow
GroupName: String
Input: String
KmsKeyArn: String
Name: String
OmitName: Boolean
PermissionsBoundary: String
RetryPolicy: RetryPolicy
RoleArn: String
ScheduleExpression: String
```



```
ScheduleExpressionTimezone: String  
StartDate: String  
State: String
```

Propiedades

DeadLetterConfig

Configure la cola del Amazon Simple Queue Service (Amazon SQS), desde la que EventBridge se envían los eventos tras una invocación de destino fallida. La invocación puede fallar, por ejemplo, cuando se envía un evento a una función de eventos de Lambda que no existe o EventBridge cuando no hay permisos suficientes para invocar la función de Lambda. Para obtener más información, consulte [Configuración de una cola de letra muerta para Scheduler en la Guía del usuario de Scheduler EventBridge](#). EventBridge

Note

El tipo de recurso [AWS::Serverless::Función](#) tiene un tipo de datos similar, `DeadLetterQueue`, que gestiona los errores que se producen tras la invocación correcta de la función de Lambda de destino. Entre los ejemplos de estos tipos de errores se incluyen la limitación de Lambda o los errores devueltos por la función de destino de Lambda. Para obtener más información sobre la *DeadLetterQueue* propiedad de la función, consulte las [Dead-letter colas](#) en la Guía para desarrolladores.AWS Lambda

Tipo: [DeadLetterConfig](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es similar a la [DeadLetterConfig](#) propiedad del tipo de `AWS::Scheduler::Schedule Target` datos. La AWS SAM versión de esta propiedad incluye subpropiedades adicionales, por si AWS SAM quiere crear automáticamente una cola de texto sin procesar.

Description

Una descripción del programa.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [Description](#) propiedad de un recurso. `AWS::Scheduler::Schedule`

EndDate

La fecha, en UTC, antes de la cual la programación puede invocar su destino. Según la expresión de recurrencia de la programación, las invocaciones se pueden detener en la EndDate que especifica o antes de ella.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [EndDate](#) propiedad de un `AWS::Scheduler::Schedule` recurso.

FlexibleTimeWindow

Permite configurar una ventana en la cual se puede invocar una programación.

Tipo: [FlexibleTimeWindow](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [FlexibleTimeWindow](#) propiedad de un `AWS::Scheduler::Schedule` recurso.

GroupName

El nombre de grupo de programación asociado a esta programación. Si no está definido, se utiliza el grupo predeterminado.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [GroupName](#) propiedad de un `AWS::Scheduler::Schedule` recurso.

Input

Texto JSON válido transmitido al destino. Si utiliza esta propiedad, no se transmitirá nada del propio texto del evento al destino.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Input](#) propiedad de un `AWS::Scheduler::Schedule Target` recurso.

KmsKeyArn

El ARN de una clave de KMS que se utiliza para cifrar datos personalizados.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [KmsKeyArn](#) propiedad de un `AWS::Scheduler::Schedule` recurso.

Name

El nombre de la programación. Si no especifica un nombre, AWS SAM genera un nombre en el formato *Function-Logical-IDEvent-Source-Name* y usa ese ID como nombre del programa.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [Name](#) propiedad de un `AWS::Scheduler::Schedule` recurso.

OmitName

De forma predeterminada, AWS SAM genera y usa un nombre de programación con el formato de *<Function-logical-ID><event-source-name>*. Establezca esta propiedad `true` para CloudFormation generar un identificador físico único y utilícelo en su lugar para el nombre del programa.

Tipo: Booleano

Obligatorio: no

Valor predeterminado: `false`

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

PermissionsBoundary

El ARN de la política que se utilizará para definir el límite de los permisos para el rol.

Note

Si `PermissionsBoundary` está definido, AWS SAM aplicará los mismos límites a la función de IAM objetivo del programador.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [PermissionsBoundary](#) propiedad de un `AWS::IAM::Role` recurso.

RetryPolicy

Objeto de la `RetryPolicy` que incluye información acerca de la configuración de la política de reintentos.

Tipo: [RetryPolicy](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [RetryPolicy](#) propiedad del tipo de `AWS::Scheduler::Schedule Target` datos.

RoleArn

El ARN de la función de IAM que EventBridge Scheduler utilizará para el destino cuando se invoque la programación.

Tipo: [RoleArn](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [RoleArn](#) propiedad del tipo de `AWS::Scheduler::Schedule Target` datos.

ScheduleExpression

La expresión de programación que determina cuándo y con qué frecuencia se ejecuta el evento programado.

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [ScheduleExpression](#) propiedad de un `AWS::Scheduler::Schedule` recurso.

ScheduleExpressionTimezone

La zona horaria en la que se evalúa la expresión de programación.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [ScheduleExpressionTimezone](#) propiedad de un `AWS::Scheduler::Schedule` recurso.

StartDate

La fecha, en UTC, después de la cual la programación puede invocar un objetivo. Según la expresión de recurrencia de la programación, las invocaciones se pueden realizar en la `StartDate` que especifica o después de ella.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [StartDate](#) propiedad de un `AWS::Scheduler::Schedule` recurso.

State

Estado de la programación del programador.

Valores aceptados `DISABLED` | `ENABLED`:

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [State](#) propiedad de un `AWS::Scheduler::Schedule` recurso.

Ejemplos

Ejemplo básico de definición de un recurso de ScheduleV2

```
Resources:
  Function:
    Properties:
      ...
    Events:
      ScheduleEvent:
        Type: ScheduleV2
        Properties:
          ScheduleExpression: "rate(1 minute)"
      ComplexScheduleEvent:
        Type: ScheduleV2
        Properties:
          ScheduleExpression: rate(1 minute)
          FlexibleTimeWindow:
            Mode: FLEXIBLE
            MaximumWindowInMinutes: 5
          StartDate: '2022-12-28T12:00:00.000Z'
          EndDate: '2023-01-28T12:00:00.000Z'
          ScheduleExpressionTimezone: UTC
          RetryPolicy:
            MaximumRetryAttempts: 5
            MaximumEventAgeInSeconds: 300
          DeadLetterConfig:
            Type: SQS
```

Note

El ID físico generado de ScheduleV2 no incluye el nombre de la pila.

SelfManagedKafka

El objeto que describe un tipo de fuente de evento SelfManagedKafka. Para obtener más información, consulte [Uso AWS Lambda con Apache Kafka autogestionado](#) en la Guía para AWS Lambda desarrolladores.

AWS Serverless Application Model (AWS SAM) genera un [AWS::Lambda::EventSourceMapping](#) recurso cuando se establece este tipo de evento.

Para utilizar Schema Registry, debe definir los permisos de rol de IAM específicos para su función. Consulte la [Configuración completa con roles de IAM](#) para ver un ejemplo de la configuración necesaria.

Sintaxis

Para declarar esta entidad en la AWS SAM plantilla, utilice la siguiente sintaxis.

YAML

```
BatchSize: Integer  
BisectBatchOnFunctionError: Boolean  
ConsumerGroupId: String  
DestinationConfig: DestinationConfig  
Enabled: Boolean  
FilterCriteria: FilterCriteria  
KafkaBootstrapServers: List  
FunctionResponseTypes: List  
KmsKeyArn: String  
LoggingConfig: LoggingConfig  
MaximumRecordAgeInSeconds: Integer  
MaximumRetryAttempts: Integer  
MetricsConfig: MetricsConfig  
ProvisionedPollerConfig: ProvisionedPollerConfig  
SchemaRegistryConfig: SchemaRegistryConfig  
SourceAccessConfigurations: SourceAccessConfigurations  
StartingPosition: String  
StartingPositionTimestamp: Double  
Topics: List
```

Propiedades

BatchSize

El número máximo de registros en cada lote que Lambda extrae del flujo o la cola y envía a su función.

Tipo: entero

Obligatorio: no

Valor predeterminado: 100

CloudFormation compatibilidad: esta propiedad se pasa directamente a la propiedad de un recurso. [BatchSize](#) AWS::Lambda::EventSourceMapping

Mínimo: 1

Máximo: 10000

BisectBatchOnFunctionError

Si la función devuelve un error, divida el lote en dos y vuelva a intentarlo.

Tipo: Booleano

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [BisectBatchOnFunctionError](#) propiedad de un AWS::Lambda::EventSourceMapping recurso.

ConsumerGroupId

Cadena que configura cómo se leerán los eventos de los temas de Kafka.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [SelfManagedKafkaEventSourceConfig](#) propiedad de un AWS::Lambda::EventSourceMapping recurso.

DestinationConfig

Un objeto de configuración que especifica el destino de un evento después de que Lambda lo procese.

Utilice esta propiedad para especificar el destino de las invocaciones fallidas desde el origen de eventos autoadministrado de Kafka.

Tipo: [DestinationConfig](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [DestinationConfig](#) propiedad de un AWS::Lambda::EventSourceMapping recurso.

Enabled

Deshabilita el mapeo de origen de eventos para pausar el sondeo y la invocación.

Tipo: Booleano

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Enabled](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

FilterCriteria

Un objeto que define los criterios que determinan si Lambda debe procesar un evento o no. Para obtener más información, consulta [Filtrado de eventos de AWS Lambda](#) en la Guía para desarrolladores de AWS Lambda .

Tipo: [FilterCriteria](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [FilterCriteria](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

KafkaBootstrapServers

La lista de servidores de arranque para sus agentes Kafka: Incluya el puerto, por ejemplo `broker.example.com:xxxx`

Tipo: lista

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

FunctionResponseTypes

Una lista de enumeraciones de tipos de respuesta actuales aplicadas a la asignación de origen de eventos. Para obtener más información, consulta [Informes de errores de artículos en lotes](#) en la Guía para desarrolladores de AWS Lambda .

Valores válidos: `ReportBatchItemFailures`

Tipo: lista

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [FunctionResponseTypes](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

KmsKeyArn

El nombre de recurso de Amazon (ARN) de la clave de que se utilizará para cifrar la información de este evento.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [KmsKeyArn](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

LoggingConfig

La configuración de registro de la fuente de eventos.

Tipo: [LoggingConfig](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [LoggingConfig](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

MaximumRecordAgeInSeconds

La antigüedad máxima de un registro que Lambda envía a una función para su procesamiento.

Tipo: entero

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [MaximumRecordAgeInSeconds](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

MetricsConfig

La configuración de métricas de la fuente de eventos.

Tipo: [MetricsConfig](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [MetricsConfig](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

MaximumRetryAttempts

El número máximo de veces que se debe volver a intentar cuando la función devuelve un error.

Tipo: entero

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [MaximumRetryAttempts](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

ProvisionedPollerConfig

Configuración para aumentar la cantidad de sondeos utilizados para calcular las asignaciones de orígenes de eventos. Esta configuración permite un mínimo de 1 sondeo y un máximo de 2000 sondeos. Para ver un ejemplo, consulte [ProvisionedPollerConfig ejemplo](#)

Tipo: [ProvisionedPollerConfig](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [ProvisionedPollerConfig](#) propiedad de un recurso.

`AWS::Lambda::EventSourceMapping`

SchemaRegistryConfig

Configuración para usar un registro de esquemas con el origen de eventos autoadministrado de Kafka.

Note

Esta característica solicita que se configure `ProvisionedPollerConfig`.

Tipo: [SchemaRegistryConfig](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [SelfManagedKafkaEventSourceConfig](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

SourceAccessConfigurations

Un conjunto del protocolo de autenticación, los componentes de la VPC y los metadatos de OAuth 2.0 que protegen y definen el origen de los eventos. Cada entrada es un objeto con un `Type` campo y, para la mayoría de los tipos, un campo `URI`. Para ver la lista completa de tipos y el valor que `URI` tiene cada uno, consulte [SourceAccessConfiguration](#) la Guía del AWS CloudFormation usuario.

AWS SAM aplica la siguiente validación antes de pasar esta propiedad a Lambda:

- No debe especificar más de un tipo de autenticación. AWS SAM devuelve un error si especifica más de uno.
- Debe especificar `VPC_SUBNET` y `VPC_SECURITY_GROUP` juntos. AWS SAM devuelve un error si especifica uno sin el otro.
- Debe especificar un `URI` valor para cada tipo excepto `IAM_AUTH` y `IAM_OAUTHBEARER_AUTH`, que acepta `noURI`. AWS SAM comprueba únicamente que el valor está presente y lo pasa sin interpretarlo. Lambda valida el valor y la combinación de tipos al realizar la implementación.

AWS SAM añade `secretsmanager:GetSecretValue` permisos para cada ARN secreto que especifique, incluido el `SERVER_ROOT_CA_CERTIFICATE` secreto, y los permisos de la interfaz de red Amazon EC2 que Lambda necesita al especificar las entradas de VPC. Para `IAM_AUTH` y `IAM_OAUTHBEARER_AUTH`, no AWS SAM agrega permisos de clúster o token. Otorgue usted mismo esos permisos para la función de ejecución de su función. Cuando ninguna de estas opciones se aplica, por ejemplo, `IAM_AUTH` sin entradas de VPC, no se AWS SAM agrega ninguna política en línea a la función de ejecución. Para conocer los permisos necesarios, consulte [Configuración de los métodos de autenticación de clústeres en Lambda](#) en la Guía AWS Lambda para desarrolladores.

Valores válidos: `BASIC_AUTH` | `CLIENT_CERTIFICATE_TLS_AUTH` | `IAM_AUTH` | `IAM_OAUTHBEARER_AUTH` | `OAUTHBEARER_AUDIENCE` | `OAUTHBEARER_AUTH` | `OAUTHBEARER_IDENTITY_POOL` | `OAUTHBEARER_LOGICAL_CLUSTER` | `OAUTHBEARER_SCOPE` | `SASL_SCRAM_256_AUTH` | `SASL_SCRAM_512_AUTH` | `SERVER_ROOT_CA_CERTIFICATE` | `VPC_SECURITY_GROUP` | `VPC_SUBNET`

Tipo: lista de [SourceAccessConfiguration](#)

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [SourceAccessConfigurations](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

StartingPosition

La posición en el flujo donde comienza la lectura.

- `AT_TIMESTAMP`: especifique el tiempo a partir del cual comenzar la lectura de registros.
- `LATEST`: registros nuevos de solo lectura.
- `TRIM_HORIZON`: procese todos los registros disponibles.

Valores válidos: `AT_TIMESTAMP` | `LATEST` | `TRIM_HORIZON`

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [StartingPosition](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

StartingPositionTimestamp

El tiempo a partir del cual comenzar la lectura, en segundos de tiempo Unix. Defina `StartingPositionTimestamp` cuando `StartingPosition` se especifica como `AT_TIMESTAMP`.

Tipo: Doble

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [StartingPositionTimestamp](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

Topics

El nombre del tema de Kafka.

Tipo: lista

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Topics](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

Ejemplos

Configuración completa con los roles de IAM

El siguiente ejemplo muestra una configuración completa que incluye la configuración del rol de IAM necesaria para usar Schema Registry:

```
Parameters:
  PreCreatedSubnetOne:
    Type: String
  PreCreatedSubnetTwo:
    Type: String

Resources:
  MyLambdaExecutionRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Version: '2012-10-17'
        Statement:
          - Action: [sts:AssumeRole]
            Effect: Allow
            Principal:
              Service: [lambda.amazonaws.com]
      Policies:
        - PolicyName: KafkaAuthPolicy
          PolicyDocument:
            Statement:
              - Action: [secretsmanager:GetSecretValue, kms:Decrypt]
                Effect: "Allow"
                Resource: ['arn:aws:secretsmanager:us-west-2:123456789012:secret:kafkaSecret-*****',
                          'arn:aws:kms:us-west-2:123456789012:key/keyId']
        - PolicyName: ENIPolicy
          PolicyDocument:
            Statement:
              - Action: [ec2:CreateNetworkInterface,
                        ec2:DescribeNetworkInterfaces, ec2:DescribeVpcs,
                        ec2>DeleteNetworkInterface,
                        ec2:DescribeSubnets, ec2:DescribeSecurityGroups]
```

```

      Effect: Allow
      Resource: '*'
    - PolicyName: SchemaRegistryPolicy
      PolicyDocument:
        Statement:
          - Action: [glue:GetRegistry]
            Effect: Allow
            Resource: 'arn:aws:glue:{region}:{account-id}:registry/{registry-name}'
    - PolicyName: SchemaVersionsPolicy
      PolicyDocument:
        Statement:
          - Action: [glue:GetSchemaVersions]
            Effect: Allow
            Resource: '*'
  ManagedPolicyArns:
    - !Sub arn:${AWS::Partition}:iam::aws:policy/service-role/
AWSLambdaBasicExecutionRole
  Tags:
    - {Value: SAM, Key: lambda:createdBy}

MyKafkaProcessor:
  Type: AWS::Serverless::Function
  Properties:
    Runtime: nodejs18.x
    Handler: index.handler
    CodeUri: ${codeuri}
    Role:
      Fn::GetAtt: [MyLambdaExecutionRole, Arn]
  Events:
    SelfManagedKafkaEvent:
      Type: SelfManagedKafka
      Properties:
        KafkaBootstrapServers:
          - my-kafka-broker-1:9092
          - my-kafka-broker-2:9092
        Topics:
          - SchemaRegistryTestTopic
        StartingPosition: LATEST
        SourceAccessConfigurations:
          - Type: VPC_SUBNET
            URI: subnet:subnet-12345678
          - Type: VPC_SECURITY_GROUP
            URI: security_group:sg-12345678
          - Type: BASIC_AUTH

```

```

      URI: !Sub arn:${AWS::Partition}:secretsmanager:us-
west-2:123456789012:secret:my-path/my-secret-name-1a2b3c
    ProvisionedPollerConfig:
      MinimumPollers: 1
    SchemaRegistryConfig:
      AccessConfigs:
        - Type: BASIC_AUTH
          URI: !Sub arn:${AWS::Partition}:secretsmanager:us-
west-2:123456789012:secret:my-path/my-secret-name-1a2b3c
      SchemaValidationConfigs:
        - Attribute: KEY
      EventRecordFormat: JSON
      SchemaRegistryURI: !Sub arn:${AWS::Partition}:glue:us-
west-2:123456789012:registry/myregistry

```

ProvisionedPollerConfig ejemplo

```

ProvisionedPollerConfig:
  MinimumPollers: 1
  MaximumPollers: 200

```

Self-managed fuente de eventos de Kafka

A continuación, se muestra un ejemplo de un tipo de fuente de evento SelfManagedKafka.

YAML

```

Events:
  SelfManagedKafkaEvent:
    Type: SelfManagedKafka
    Properties:
      BatchSize: 1000
      Enabled: true
      KafkaBootstrapServers:
        - abc.xyz.com:xxxx
      SourceAccessConfigurations:
        - Type: BASIC_AUTH
          URI: arn:aws:secretsmanager:us-west-2:123456789012:secret:my-path/my-secret-
name-1a2b3c
      Topics:
        - MyKafkaTopic

```


Self-managed Fuente de eventos de Kafka con autenticación OAuth 2.0

En el siguiente ejemplo, se autentica con SASL/OAUTHBEARER. La OAUTHBEARER_AUTH entrada apunta al secreto que contiene las credenciales de OAuth 2.0, y cada entrada de metadatos contiene un valor literal que Lambda pasa a su proveedor o agente de identidad.

```
Events:
  SelfManagedKafkaEvent:
    Type: SelfManagedKafka
    Properties:
      KafkaBootstrapServers:
        - abc.xyz.com:9092
      Topics:
        - MyKafkaTopic
      StartingPosition: LATEST
      ConsumerGroupId: consumergroup1
      SourceAccessConfigurations:
        - Type: OAUTHBEARER_AUTH
          URI: arn:aws:secretsmanager:us-west-2:123456789012:secret:my-path/my-oauth-secret-1a2b3c
        - Type: OAUTHBEARER_SCOPE
          URI: read write
        - Type: OAUTHBEARER_AUDIENCE
          URI: https://api.example.com
        - Type: SERVER_ROOT_CA_CERTIFICATE
          URI: arn:aws:secretsmanager:us-west-2:123456789012:secret:my-path/my-server-root-ca-cert-7p8q9r
        - Type: VPC_SUBNET
          URI: subnet:subnet-12345678
        - Type: VPC_SECURITY_GROUP
          URI: security_group:sg-12345678
```

Self-managed Fuente de eventos de Kafka con autenticación de IAM

En el siguiente ejemplo, se autentica con IAM. La IAM_AUTH entrada no ocupa ningún URI campo porque Lambda firma cada conexión con las credenciales del rol de ejecución de la función. Otorgue a su rol de ejecución los permisos de clúster que Lambda requiere, ya que AWS SAM no los agrega.

```
Events:
  SelfManagedKafkaEvent:
    Type: SelfManagedKafka
    Properties:
```

```
KafkaBootstrapServers:
  - abc.xyz.com:9092
Topics:
  - MyKafkaTopic
StartingPosition: LATEST
ConsumerGroupId: consumergroup1
SourceAccessConfigurations:
  - Type: IAM_AUTH
  - Type: VPC_SUBNET
    URI: subnet:subnet-12345678
  - Type: VPC_SECURITY_GROUP
    URI: security_group:sg-12345678
```

Self-managed Fuente de eventos de Kafka con autenticación de IAM y OAuth 2.0

En el siguiente ejemplo, se autentica con. `IAM_OAUTHBEARER_AUTH` Esta entrada no ocupa ningún URI campo, pero debe emparejarla con una `OAUTHBEARER_AUDIENCE` entrada que URI sea la audiencia que esperan sus corredores. Lambda solicita un token de identidad web de corta duración para la función de ejecución, que necesita el `sts:GetWebIdentityToken` permiso, y sus agentes deben confiar en el AWS emisor y aceptar el valor de audiencia.

```
Events:
  SelfManagedKafkaEvent:
    Type: SelfManagedKafka
    Properties:
      KafkaBootstrapServers:
        - abc.xyz.com:9092
      Topics:
        - MyKafkaTopic
      StartingPosition: LATEST
      ConsumerGroupId: consumergroup1
      SourceAccessConfigurations:
        - Type: IAM_OAUTHBEARER_AUTH
        - Type: OAUTHBEARER_AUDIENCE
          URI: kafka-cluster
        - Type: VPC_SUBNET
          URI: subnet:subnet-12345678
        - Type: VPC_SECURITY_GROUP
          URI: security_group:sg-12345678
```

Self-managed Kafka Event Source con AWS Glue Schema Registry

El siguiente es un ejemplo de un tipo de fuente de SelfManagedKafka eventos configurado con AWS Glue Schema Registry.

```
Events:
  SelfManagedKafkaEvent:
    Type: SelfManagedKafka
    Properties:
      KafkaBootstrapServers:
        - abc.xyz.com:9092
      Topics:
        - SchemaRegistryTestTopic
      StartingPosition: LATEST
      ProvisionedPollerConfig:
        MinimumPollers: 1
      SchemaRegistryConfig:
        SchemaRegistryURI: !Sub arn:${AWS::Partition}:glue:us-
west-2:123456789012:registry/myregistry
        EventRecordFormat: JSON
        SchemaValidationConfigs:
          - Attribute: KEY
          - Attribute: VALUE
      SourceAccessConfigurations:
        - Type: VPC_SUBNET
          URI: subnet:subnet-12345678
        - Type: VPC_SECURITY_GROUP
          URI: security_group:sg-12345678
```

Self-managed Fuente de eventos de Kafka con registro de esquemas confluentes

A continuación, se muestra un ejemplo de un tipo de origen de evento de SelfManagedKafka configurado con Schema Registry confluyente.

```
Events:
  SelfManagedKafkaEvent:
    Type: SelfManagedKafka
    Properties:
      KafkaBootstrapServers:
        - abc.xyz.com:9092
      Topics:
        - SchemaRegistryTestTopic
      StartingPosition: LATEST
```

```

ProvisionedPollerConfig:
  MinimumPollers: 1
SchemaRegistryConfig:
  SchemaRegistryURI: https://my-schema-registry.confluent.cloud
  AccessConfigs:
    - Type: BASIC_AUTH
      URI: !Sub arn:${AWS::Partition}:secretsmanager:us-
west-2:123456789012:secret:my-secret
  EventRecordFormat: JSON
  SchemaValidationConfigs:
    - Attribute: KEY
    - Attribute: VALUE
SourceAccessConfigurations:
  - Type: VPC_SUBNET
    URI: subnet:subnet-12345678
  - Type: VPC_SECURITY_GROUP
    URI: security_group:sg-12345678
  - Type: BASIC_AUTH
    URI: !Sub arn:${AWS::Partition}:secretsmanager:us-
west-2:123456789012:secret:kafka-secret

```

SNS

El objeto que describe un tipo de fuente de evento de SNS.

SAM genera recursos de [AWS::SNS::Subscription](#) cuando se establece este tipo de evento

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```

FilterPolicy: SnsFilterPolicy
FilterPolicyScope: String
RedrivePolicy: Json
Region: String
SqsSubscription: Boolean | SqsSubscriptionObject
Topic: String

```

Propiedades

FilterPolicy

El JSON de la política de filtro asignada a la suscripción. Para obtener más información, consulte la referencia [GetSubscriptionAttributes](#) de la API de Amazon Simple Notification Service.

Tipo: [SnsFilterPolicy](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [FilterPolicy](#) propiedad de un `AWS::SNS::Subscription` recurso.

FilterPolicyScope

Este atributo permite elegir el alcance de filtrado mediante uno de los siguientes tipos de valores de cadena:

- `MessageAttributes`: el filtro se aplica a los atributos del mensaje.
- `MessageBody`: el filtro se aplica al cuerpo del mensaje.

Tipo: cadena

Obligatorio: no

Valor predeterminado: `MessageAttributes`

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [FilterPolicyScope](#) propiedad de un `AWS::SNS::Subscription` recurso.

RedrivePolicy

Cuando se especifica, envía mensajes que no se pueden entregar en la cola de mensajes sin procesar de Amazon SQS especificada. Los mensajes que no se pueden entregar debido a errores del cliente (por ejemplo, cuando no se puede acceder al punto de conexión) o errores del servidor (por ejemplo, cuando el servicio que alimenta el punto de conexión suscrito deja de estar disponible) se mantienen en la cola de mensajes fallidos para su posterior análisis o reprocesamiento.

Para obtener más información sobre la política de redireccionamiento y las colas de mensajes fallidos, consulta [Uso de colas de mensajes fallidos en Amazon SQS](#) en la Guía para desarrolladores de Amazon Simple Queue Service.

Tipo: Json

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [RedrivePolicy](#) propiedad de un `AWS::SNS::Subscription` recurso.

Region

Para suscripciones entre regiones, la región en la que reside el tema.

Si no se especifica ninguna región, CloudFormation utiliza la región de la persona que llama como predeterminada.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [Region](#) propiedad de un `AWS::SNS::Subscription` recurso.

SqsSubscription

Establezca esta propiedad en `true` o especifique `SqsSubscriptionObject` para habilitar la agrupación por lotes de notificaciones de temas de SNS en una cola de SQS. Si se establece esta propiedad para que `true` cree una nueva cola de SQS, mientras si se especifica `SqsSubscriptionObject` utiliza una cola de SQS existente.

Tipo: booleano | [SqsSubscriptionObject](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

Topic

El ARN del tema al que se suscribe.

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [TopicArn](#) propiedad de un `AWS::SNS::Subscription` recurso.

Ejemplos

Ejemplo de orígenes de eventos de SNS

Ejemplo de orígenes de eventos de SNS

YAML

```
Events:
  SNSEvent:
    Type: SNS
    Properties:
      Topic: arn:aws:sns:us-east-1:123456789012:my_topic
      SqsSubscription: true
      FilterPolicy:
        store:
          - example_corp
        price_usd:
          - numeric:
              - ">="
              - 100
```

SqsSubscriptionObject

Especifica una opción de cola de SQS existente para el evento de SNS

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
BatchSize: String
Enabled: Boolean
QueueArn: String
QueuePolicyLogicalId: String
QueueUrl: String
```

Propiedades

BatchSize

El número máximo de elementos a recuperar en un único lote para la cola SQS.

Tipo: cadena

Obligatorio: no

Valor predeterminado: 10

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

Enabled

Deshabilita la asignación de orígenes de eventos SQS para pausar el sondeo y la invocación.

Tipo: Booleano

Obligatorio: no

Valor predeterminado: True

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

QueueArn

Especifica un arn de cola de SQS existente.

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

QueuePolicyLogicalId

Proporcione un nombre LogicalId personalizado para el recurso. [AWS::SQS::QueuePolicy](#)

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un equivalente.
CloudFormation

QueueUrl

Especifica la URL de la cola asociada a la propiedad QueueArn.

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un
CloudFormation equivalente.

Ejemplos

Evento SQS para SNS existente

Ejemplo de añadir una cola de SQS existente para suscribirse a un tema de SNS.

YAML

```
QueuePolicyLogicalId: CustomQueuePolicyLogicalId
QueueArn:
  Fn::GetAtt: MyCustomQueue.Arn
QueueUrl:
  Ref: MyCustomQueue
BatchSize: 5
```

SQS

El objeto que describe un tipo de fuente de evento de SQS. Para obtener más información, consulte [Uso AWS Lambda con Amazon SQS en la Guía para AWS Lambda](#) desarrolladores.

SAM genera recursos de [AWS::Lambda::EventSourceMapping](#) cuando se establece este tipo de evento

Sintaxis

Para declarar esta entidad en su plantilla AWS Serverless Application Model (AWS SAM), utilice la siguiente sintaxis.

YAML

```
BatchSize: Integer  
Enabled: Boolean  
FilterCriteria: FilterCriteria  
FunctionResponseTypes: List  
KmsKeyArn: String  
MaximumBatchingWindowInSeconds: Integer  
MetricsConfig: MetricsConfig  
ProvisionedPollerConfig: ProvisionedPollerConfig  
Queue: String  
ScalingConfig: ScalingConfig
```

Propiedades

BatchSize

El número máximo de elementos que se recuperan en un único lote.

Tipo: entero

Obligatorio: no

Valor predeterminado: 10

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [BatchSize](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

Mínimo: 1

Máximo: 10000

Enabled

Deshabilita el mapeo de origen de eventos para pausar el sondeo y la invocación.

Tipo: Booleano

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Enabled](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

FilterCriteria

Un objeto que define los criterios que determinan si Lambda debe procesar un evento o no. Para obtener más información, consulta [Filtrado de eventos de AWS Lambda](#) en la Guía para desarrolladores de AWS Lambda .

Tipo: [FilterCriteria](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [FilterCriteria](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

FunctionResponseTypes

Una lista de enumeraciones de tipos de respuesta actuales aplicadas a la asignación de origen de eventos. Para obtener más información, consulta [Informes de errores de artículos en lotes](#) en la Guía para desarrolladores de AWS Lambda .

Valores válidos: `ReportBatchItemFailures`

Tipo: lista

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [FunctionResponseTypes](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

KmsKeyArn

El nombre de recurso de Amazon (ARN) de la clave de que se utilizará para cifrar la información de este evento.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [KmsKeyArn](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

MaximumBatchingWindowInSeconds

La cantidad de tiempo máxima, en segundos, para recopilar registros antes de invocar la función.

Tipo: entero

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [MaximumBatchingWindowInSeconds](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

MetricsConfig

Una configuración opcional para obtener métricas mejoradas para las asignaciones de orígenes de eventos que capturan cada etapa del procesamiento. Para ver un ejemplo, consulta [MetricsConfig evento](#).

Tipo: [MetricsConfig](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [MetricsConfig](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

ProvisionedPollerConfig

Configuración para aumentar la cantidad de sondeos utilizados para calcular las asignaciones de orígenes de eventos. Esta configuración permite un mínimo de 2 sondeadores y un máximo de 2000 sondeos. Para ver un ejemplo, consulte [ProvisionedPollerConfig ejemplo](#).

Tipo: [ProvisionedPollerConfig](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [ProvisionedPollerConfig](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

Queue

El ARN de la cola.

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [EventSourceArn](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

ScalingConfig

Configuración del escalado de los pollers SQS para controlar la tasa de invocaciones y establecer el máximo de invocaciones simultáneas.

Tipo: [ScalingConfig](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [ScalingConfig](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

Ejemplos

MetricsConfig evento

El siguiente ejemplo muestra un recurso que usa la propiedad `MetricsConfig` para capturar cada etapa del procesamiento para sus asignaciones de orígenes de eventos.

```
Resources:
  FilteredEventsFunction:
    Type: AWS::Serverless::Function
    Properties:
      CodeUri: s3://sam-demo-bucket/metricsConfig.zip
      Handler: index.handler
      Runtime: nodejs16.x
      Events:
        KinesisStream:
          Type: Kinesis
          Properties:
            Stream: !GetAtt KinesisStream.Arn
            StartingPosition: LATEST
            MetricsConfig:
              Metrics:
                - EventCount
```

Evento de SQS básico

```
Events:
  SQSEvent:
    Type: SQS
    Properties:
```

```

Queue: arn:aws:sqs:us-west-2:012345678901:my-queue
BatchSize: 10
Enabled: false
FilterCriteria:
  Filters:
    - Pattern: '{"key": ["val1", "val2"]}'

```

Configura los informes por lotes parciales para su cola de SQS

```

Events:
  SQSEvent:
    Type: SQS
    Properties:
      Enabled: true
      FunctionResponseTypes:
        - ReportBatchItemFailures
      Queue: !GetAtt MySqsQueue.Arn
      BatchSize: 10

```

Función de eventos Lambda con un evento SQS que tiene el escalado configurado

```

MyFunction:
  Type: AWS::Serverless::Function
  Properties:
    ...
    Events:
      MySQSEvent:
        Type: SQS
        Properties:
          ...
          ScalingConfig:
            MaximumConcurrency: 10

```

ProvisionedPollerConfig ejemplo

```

MyFunction:
  Type: AWS::Serverless::Function
  Properties:
    Handler: index.handler
    Runtime: nodejs18.x
    Timeout: 30
    Events:

```

```
SQSEvent:
  Type: SQS
  Properties:
    Queue: !GetAtt MyQueue.Arn
    BatchSize: 10
    Enabled: True
    ProvisionedPollerConfig:
      MaximumPollers: 300
      MinimumPollers: 10
```

FunctionCode

El [paquete de despliegue](#) de una función de eventos Lambda.

Sintaxis

Para declarar esta entidad en su plantilla AWS Serverless Application Model (AWS SAM), utilice la siguiente sintaxis.

YAML

```
Bucket: String
Key: String
StorageMode: String
Version: String
```

Propiedades

Bucket

Un bucket de Amazon S3 en la misma AWS región que su función.

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [S3Bucket](#) propiedad del tipo de `AWS::Lambda::Function` Code datos.

Key

La clave de Amazon S3 del paquete de implementación.

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [S3Key](#) propiedad del tipo de `AWS::Lambda::Function` Code datos.

StorageMode

Controla la forma en que Lambda almacena el paquete de despliegue.

- **COPY**— Lambda carga una copia del paquete de implementación en el almacenamiento. Lambda-managed Puede eliminar el objeto de Amazon S3 después de que Lambda haya creado la función.
- **REFERENCE**— Lambda hace referencia al paquete de despliegue de su bucket de Amazon S3 y no almacena ninguna copia. El objeto debe permanecer en su lugar y Lambda debe mantener el acceso a él durante toda la vida útil de la función.

Para usarlo **REFERENCE**, habilite el control de versiones en su bucket de Amazon S3 y conceda al servicio Lambda el acceso principal al objeto. Para obtener más información, consulte [Almacenamiento de código de Self-managed Amazon S3](#) en la Guía para AWS Lambda desarrolladores.

Tipo: cadena

Valores válidos: COPY | REFERENCE

Obligatorio: no

Valor predeterminado: COPY

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [S3ObjectStorageMode](#) propiedad del tipo de `AWS::Lambda::Function` Code datos.

Version

Para objetos con control de versiones, la versión del objeto del paquete de implementación que se va a utilizar.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [S3ObjectVersion](#) propiedad del tipo de `AWS::Lambda::Function` Code datos.

Ejemplos

FunctionCode

CodeUri: código de la función de ejemplo

YAML

```
CodeUri:
  Bucket: sam-s3-demo-bucket-name
  Key: mykey-name
  Version: 121212
```

Self-managed Almacenamiento de código en Amazon S3

Lo siguiente **CodeUri** hace referencia al paquete de despliegue de su propio bucket de Amazon S3 en lugar de copiarlo en el Lambda-managed almacenamiento.

YAML

```
CodeUri:
  Bucket: amzn-s3-demo-bucket-name
  Key: mykey-name
  Version: 121212
  StorageMode: REFERENCE
```

FunctionUrlConfig

Crea una URL de AWS Lambda función con los parámetros de configuración especificados. Una URL de función de Lambda es un punto de conexión HTTP(S) dedicado que puede utilizar para invocar la función.

De forma predeterminada, la URL de la función que cree utiliza la \$LATEST versión de su función de eventos Lambda. Si especifica un **AutoPublishAlias** para la función de Lambda, el punto de conexión se conecta al alias de la función especificada.

Para obtener más información, consulte [Errores de la función de Lambda](#) en la Guía para desarrolladores de AWS Lambda .

Sintaxis

Para declarar esta entidad en su plantilla AWS Serverless Application Model (AWS SAM), utilice la siguiente sintaxis.

YAML

```
AuthType: String
Cors: Cors
InvokeMode: String
```

Propiedades

AuthType

El tipo de autenticación que utiliza la URL de función. Para usar AWS Identity and Access Management (IAM) para autorizar solicitudes, establézcalo en `AWS_IAM`. Para un acceso abierto, establézcalo en `NONE`.

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [AuthType](#) propiedad de un `AWS::Lambda::Url` recurso.

Cors

La configuración de uso compartido de recursos entre orígenes (CORS) para la URL de función.

Tipo: [Cors](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Cors](#) propiedad de un `AWS::Lambda::Url` recurso.

InvokeMode

El modo en el que se invocará la URL de función. Para que la función devuelva la respuesta una vez finalizada la invocación, establézcala en `BUFFERED`. Para que su función transmita la respuesta, establézcala en `RESPONSE_STREAM`. El valor predeterminado es `BUFFERED`.

Valores válidos: `BUFFERED` o `RESPONSE_STREAM`

Tipo: cadena

Obligatorio: no

AWS CloudFormation compatibilidad: esta propiedad se pasa directamente a la [InvokeMode](#) propiedad de un `AWS::Lambda::Url` recurso.

Ejemplos

URL de función

El siguiente ejemplo crea una función de Lambda con una URL de función. La URL de función utiliza la autorización de IAM.

YAML

```
HelloWorldFunction:
  Type: AWS::Serverless::Function
  Properties:
    CodeUri: hello_world/
    Handler: index.handler
    Runtime: nodejs20.x
    FunctionUrlConfig:
      AuthType: AWS_IAM
      InvokeMode: RESPONSE_STREAM

Outputs:
  MyFunctionUrlEndpoint:
    Description: "My Lambda Function URL Endpoint"
    Value:
      Fn::GetAtt: HelloWorldFunctionUrl.FunctionUrl
```

CapacityProviderConfig

Configura el proveedor de capacidad al que se adjuntarán las versiones publicadas de la función. Esto permite que la función se ejecute en EC2 instancias propiedad del cliente administradas por Lambda.

Note

Esta configuración determina el tipo de procesamiento de una función y debe especificarse durante la primera implementación de la función. No se puede añadir ni eliminar una vez creado el recurso de la función.

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
Arn: String  
ExecutionEnvironmentMemoryGiBPerVCpu: Float  
PerExecutionEnvironmentMaxConcurrency: Integer
```

Propiedades

Arn

El ARN del proveedor de capacidad que se utilizará para esta función.

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad es exclusiva de SAM.

ExecutionEnvironmentMemoryGiBPerVCpu

La relación entre memoria (en GiB) y vCPU para cada entorno de ejecución.

Note

La relación de memoria por CPU no puede superar la memoria total de la función, que es de 2048 MB. memory-to-CPU Las proporciones admitidas son de 2 GB, 4 GB u 8 GB por CPU.

Tipo: flotante

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [ExecutionEnvironmentMemoryGiBPerVCpu](#) propiedad de un `AWS::Lambda::Function` recurso.

PerExecutionEnvironmentMaxConcurrency

El número máximo de ejecuciones simultáneas por entorno de ejecución (sandbox).

Tipo: entero

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [PerExecutionEnvironmentMaxConcurrency](#) propiedad de un `AWS::Lambda::Function` recurso.

Ejemplos

Configuración de los proveedores de capacidad

El siguiente ejemplo muestra una configuración de proveedor de capacidad que hace referencia a un recurso del proveedor de capacidad.

```
CapacityProviderConfig:
  Arn: !GetAtt MyCapacityProvider.Arn
  ExecutionEnvironmentMemoryGiBPerVCpu: 4.0
  PerExecutionEnvironmentMaxConcurrency: 100
```

FunctionScalingConfig

Configura el comportamiento de escalado de las versiones de la Función de Evento Lambda, controlando el número de entornos de ejecución (sandboxes) que se pueden crear. Esta configuración se aplica a las versiones de funciones numéricas LATEST.PUBLISHED y \$.

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
MinExecutionEnvironments: Integer
MaxExecutionEnvironments: Integer
```

Propiedades

MinExecutionEnvironments

El número mínimo de entornos de ejecución que se deben mantener para la versión de la función.

Tipo: entero

Obligatorio: no

Valor predeterminado: 3

Mínimo: 0

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [MinExecutionEnvironments](#) propiedad de un `AWS::Lambda::Function` recurso.

MaxExecutionEnvironments

El número máximo de entornos de ejecución que se pueden crear para la versión de la función.

Tipo: entero

Obligatorio: no

Valor predeterminado: 3

Mínimo: 0

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [MaxExecutionEnvironments](#) propiedad de un `AWS::Lambda::Function` recurso.

Ejemplos

Configuración de escalado de funciones

El siguiente ejemplo muestra una configuración de escalado de funciones con entornos de ejecución mínima y máxima.

```
FunctionScalingConfig:
  MinExecutionEnvironments: 5
  MaxExecutionEnvironments: 100
```

AWS::Serverless:: API GraphQL

Usa el tipo de `AWS::Serverless::GraphQLApi` recurso AWS Serverless Application Model (AWS SAM) para crear y configurar una AWS AppSync GraphQL API para tu aplicación sin servidor.

Para obtener más información AWS AppSync, consulte [¿Qué es? AWS AppSync](#) en la Guía para AWS AppSync desarrolladores.

Sintaxis

YAML

LogicalId:

Type: `AWS::Serverless::GraphQLApi`

Properties:

ApiKeys: [ApiKeys](#)

Auth: [Auth](#)

Cache: [AWS::AppSync::ApiCache](#)

DataSources: [DataSource](#)

DomainName: [AWS::AppSync::DomainName](#)

Functions: [Function](#)

Logging: [LogConfig](#)

Name: *String*

Resolvers: [Resolver](#)

SchemaInline: *String*

SchemaUri: *String*

Tags:

- [Tag](#)

XrayEnabled: *Boolean*

Propiedades

ApiKeys

Cree una clave única que se pueda utilizar para realizar operaciones en GraphQL que requieran una clave de API.

Tipo: [ApiKeys](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un equivalente. CloudFormation

Auth

Configure la autenticación para tu GraphQL API.

Tipo: [Auth](#)

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

Cache

La entrada de una operación CreateApiCache.

Tipo: [AWS::AppSync:: ApiCache](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente al ApiCache recurso [AWS::AppSync::](#).

DataSources

Cree fuentes de datos para las funciones AWS AppSync a las que conectarse. AWS SAM es compatible con Amazon AWS Lambda DynamoDB y las fuentes de datos.

Tipo: [DataSource](#)

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

DomainName

Es el nombre de dominio personalizado para la API GraphQL.

Tipo: [AWS::AppSync:: DomainName](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente al DomainName recurso [AWS::AppSync::](#). AWS SAM genera automáticamente el DomainNameApiAssociation recurso [AWS::AppSync::](#).

Functions

Configure funciones en las API de GraphQL para realizar ciertas operaciones.

Tipo : [Función](#)

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

Logging

Configura el CloudWatch registro de Amazon para tu GraphQL API.

Si no especificas esta propiedad, AWS SAM generará `CloudWatchLogsRoleArn` y establecerá los siguientes valores:

- `ExcludeVerboseContent`: `true`
- `FieldLogLevel`: `ALL`

Para excluirse del registro, especifica lo siguiente:

```
Logging: false
```

Tipo: [LogConfig](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [LogConfig](#) propiedad de un `AWS::AppSync::GraphQLApi` recurso.

LogicalId

El nombre exclusivo de la API GraphQL de.

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Name](#) propiedad de un `AWS::AppSync::GraphQLApi` recurso.

Name

El nombre de la API GraphQL. Especifica esta propiedad para anular el valor `LogicalId`.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Name](#) propiedad de un `AWS::AppSync::GraphQLApi` recurso.

Resolvers

Configure los solucionadores para los campos de la API GraphQL. AWS SAM admite [solucionadores de JavaScript canalizaciones](#).

Tipo: Solucionador

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

SchemaInline

La representación de texto de un esquema de GraphQL en formato SDL.

Tipo: cadena

Obligatorio: condicional. Debe especificar `SchemaInline` o `SchemaUri`.

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Definition](#) propiedad de un `AWS::AppSync::GraphQLSchema` recurso.

SchemaUri

El URI del bucket de Amazon Simple Storage Service (Amazon S3) o la ruta de acceso a un directorio local.

Si especifica una ruta a una carpeta local, CloudFormation requiere que el archivo se cargue primero en Amazon S3 antes de la implementación. Puedes utilizar la AWS SAM CLI para facilitar este proceso. Para obtener más información, consulta [Cómo AWS SAM carga archivos locales en el momento de la implementación](#).

Tipo: cadena

Obligatorio: condicional. Debe especificar `SchemaInline` o `SchemaUri`.

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [DefinitionS3Location](#) propiedad de un `AWS::AppSync::GraphQLSchema` recurso.

Tags

Etiquetas (pares de clave-valor) de esta API GraphQL. Utiliza etiquetas para identificar y clasificar los recursos.

Tipo: lista de [etiqueta](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Tag](#) propiedad de un `AWS::AppSync::GraphQLApi` recurso.

XrayEnabled

Indique si desea utilizar el [AWS X-Ray rastreo](#) para este recurso.

Tipo: Booleano

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [XrayEnabled](#) propiedad de un `AWS::AppSync::GraphQLApi` recurso.

valores devueltos

Para obtener una lista de los valores devueltos, consulte [AWS::Serverless::GraphQLApi](#) en la [Guía del usuario de CloudFormation](#).

Ejemplos

API GraphQL con fuente de datos de DynamoDB

En este ejemplo, creamos una API GraphQL que usa una tabla de DynamoDB como origen de datos.

`schema.graphql`

```
schema {  
  query: Query  
  mutation: Mutation  
}  
  
type Query {  
  getPost(id: String!): Post  
}  
  
type Mutation {  
  addPost(author: String!, title: String!, content: String!): Post!  
}  
  
type Post {  
  id: String!  
  author: String  
  title: String  
  content: String  
  ups: Int!  
  downs: Int!  
  version: Int!  
}
```

template.yaml

```
AWS::Serverless::API GraphQL  
AWSTemplateFormatVersion: '2010-09-09'  
Transform: AWS::Serverless-2016-10-31  
...  
Resources:  
  DynamoDBPostsTable:  
    Type: AWS::Serverless::SimpleTable  
  
  MyGraphQLAPI:  
    Type: AWS::Serverless::GraphQLApi  
    Properties:  
      SchemaUri: ./sam_graphql_api/schema.graphql  
      Auth:  
        Type: AWS_IAM  
      DataSources:  
        DynamoDb:  
          PostsDataSource:  
            TableName: !Ref DynamoDBPostsTable  
            TableArn: !GetAtt DynamoDBPostsTable.Arn
```

```

Functions:
  preprocessPostItem:
    Runtime:
      Name: APPSYNC_JS
      Version: 1.0.0
    DataSource: NONE
    CodeUri: ./sam_graphql_api/preprocessPostItem.js
  createPostItem:
    Runtime:
      Name: APPSYNC_JS
      Version: "1.0.0"
    DataSource: PostsDataSource
    CodeUri: ./sam_graphql_api/createPostItem.js
  getPostFromTable:
    Runtime:
      Name: APPSYNC_JS
      Version: "1.0.0"
    DataSource: PostsDataSource
    CodeUri: ./sam_graphql_api/getPostFromTable.js
Resolvers:
  Mutation:
    addPost:
      Runtime:
        Name: APPSYNC_JS
        Version: "1.0.0"
      Pipeline:
        - preprocessPostItem
        - createPostItem
  Query:
    getPost:
      CodeUri: ./sam_graphql_api/getPost.js
      Runtime:
        Name: APPSYNC_JS
        Version: "1.0.0"
      Pipeline:
        - getPostFromTable

```

createPostItem.js

```

import { util } from "@aws-appsync/utils";

export function request(ctx) {
  const { key, values } = ctx.prev.result;

```

```
    return {
      operation: "PutItem",
      key: util.dynamodb.toMapValues(key),
      attributeValues: util.dynamodb.toMapValues(values),
    };
  }

  export function response(ctx) {
    return ctx.result;
  }
```

getPostFromTable.js

```
import { util } from "@aws-appsync/utils";

export function request(ctx) {
  return dynamoDBGetItemRequest({ id: ctx.args.id });
}

export function response(ctx) {
  return ctx.result;
}

/**
 * A helper function to get a DynamoDB item
 */
function dynamoDBGetItemRequest(key) {
  return {
    operation: "GetItem",
    key: util.dynamodb.toMapValues(key),
  };
}
```

preproceso PostItem.js

```
import { util } from "@aws-appsync/utils";

export function request(ctx) {
  const id = util.autoId();
  const { ...values } = ctx.args;
  values.upsert = 1;
  values.downs = 0;
  values.version = 1;
```

```
    return { payload: { key: { id }, values: values } };
  }

  export function response(ctx) {
    return ctx.result;
  }
```

Este es nuestro código de resolución:

getPost.js

```
export function request(ctx) {
  return {};
}

export function response(ctx) {
  return ctx.prev.result;
}
```

GraphQL API con una función de eventos Lambda como fuente de datos

En este ejemplo, creamos una GraphQL API que utiliza una función de eventos de Lambda como fuente de datos.

template.yaml

```
AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31
...
Resources:
  MyLambdaFunction:
    Type: AWS::Serverless::Function
    Properties:
      Handler: index.handler
      Runtime: nodejs20.x
      CodeUri: ./lambda

  MyGraphQLAPI:
    Type: AWS::Serverless::GraphQLApi
    Properties:
      Name: MyApi
      SchemaUri: ./gql/schema.gql
      Auth:
```

```

    Type: API_KEY
  ApiKeys:
    MyApiKey:
      Description: my api key
  DataSources:
    Lambda:
      MyLambdaDataSource:
        FunctionArn: !GetAtt MyLambdaFunction.Arn
  Functions:
    lambdaInvoker:
      Runtime:
        Name: APPSYNC_JS
        Version: 1.0.0
      DataSource: MyLambdaDataSource
      CodeUri: ./gql/invoke.js
  Resolvers:
    Mutation:
      addPost:
        Runtime:
          Name: APPSYNC_JS
          Version: 1.0.0
        Pipeline:
          - lambdaInvoker
    Query:
      getPost:
        Runtime:
          Name: APPSYNC_JS
          Version: 1.0.0
        Pipeline:
          - lambdaInvoker

```

```

Outputs:
  MyGraphQLAPI:
    Description: AppSync API
    Value: !GetAtt MyGraphQLAPI.GraphQLUrl
  MyGraphQLAPIMyApiKey:
    Description: API Key for authentication
    Value: !GetAtt MyGraphQLAPIMyApiKey.ApiKey

```

schema.graphql

```

schema {
  query: Query

```



```
mutation: Mutation
}
type Query {
  getPost(id: ID!): Post
}
type Mutation {
  addPost(id: ID!, author: String!, title: String, content: String): Post!
}
type Post {
  id: ID!
  author: String!
  title: String
  content: String
  ups: Int
  downs: Int
}
```

Estas son nuestras funciones:

lambda/index.js

```
exports.handler = async (event) => {
  console.log("Received event {}", JSON.stringify(event, 3));

  const posts = {
    1: {
      id: "1",
      title: "First book",
      author: "Author1",
      content: "Book 1 has this content",
      ups: "100",
      downs: "10",
    },
  };

  console.log("Got an Invoke Request.");
  let result;
  switch (event.field) {
    case "getPost":
      return posts[event.arguments.id];
    case "addPost":
      // return the arguments back
      return event.arguments;
    default:
```

```
    throw new Error("Unknown field, unable to resolve " + event.field);  
  }  
};
```

invoker.js

```
import { util } from "@aws-appsync/utils";  
  
export function request(ctx) {  
  const { source, args } = ctx;  
  return {  
    operation: "Invoke",  
    payload: { field: ctx.info.fieldName, arguments: args, source },  
  };  
}  
  
export function response(ctx) {  
  return ctx.result;  
}
```

ApiKeys

Cree una clave única que se pueda utilizar para realizar operaciones en GraphQL que requieran una clave de API.

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
LogicalId:  
  ApiKeyId: String  
  Description: String  
  ExpiresOn: Double
```

Propiedades

ApiKeyId

El nombre único de su clave de API. Especifica si desea anular el valor de LogicalId.

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [ApiKeyId](#) propiedad de un `AWS::AppSync::ApiKey` recurso.

Description

Descripción de su clave de API.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Description](#) propiedad de un `AWS::AppSync::ApiKey` recurso.

ExpiresOn

El tiempo después del cual caduca la clave de API. La fecha se representa como segundos desde la marca temporal Epoch, redondeados a la hora más cercana.

Tipo: Doble

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Expires](#) propiedad de un `AWS::AppSync::ApiKey` recurso.

LogicalId

El nombre único de su clave de API.

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [ApiKeyId](#) propiedad de un `AWS::AppSync::ApiKey` recurso.

Auth

Configura la autorización para su API GraphQL.

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
Additional:
- AuthProvider
LambdaAuthorizer: LambdaAuthorizerConfig
OpenIDConnect: OpenIDConnectConfig
Type: String
UserPool: UserPoolConfig
```

Propiedades

Additional

Una lista de tipos de autorización adicionales para su API GraphQL.

Tipo: Lista de [AuthProvider](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

LambdaAuthorizer

Especifique la configuración de autorización opcional para su autorizador de funciones de eventos Lambda. Puedes configurar esta propiedad opcional si Type se especifica como AWS_LAMBDA.

Tipo: [LambdaAuthorizerConfig](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [LambdaAuthorizerConfig](#) propiedad de un AWS::AppSync::GraphQLApi recurso.

OpenIDConnect

Especifica la configuración de autorización opcional para su servicio OpenID Connect compatible. Puedes configurar esta propiedad opcional si Type se especifica como OPENID_CONNECT.

Tipo: [OpenIDConnectConfig](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [OpenIDConnectConfig](#) propiedad de un `AWS::AppSync::GraphQLApi` recurso.

Type

El tipo de autorización predeterminado entre las aplicaciones y tu AWS AppSync GraphQL API.

Para obtener una lista y una descripción de los valores permitidos, consulta [Autorización y autenticación](#) en la Guía para desarrolladores de AWS AppSync .

Al especificar un autorizador de Lambda (`AWS_LAMBDA`), AWS SAM crea una AWS Identity and Access Management política (de IAM) para aprovisionar permisos entre la API GraphQL y la función de Lambda.

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [AuthenticationType](#) propiedad de un recurso `AWS::AppSync::GraphQLApi`.

UserPool

Especifica la configuración de autorización opcional para utilizar grupos de usuarios de Amazon Cognito. Puedes configurar esta propiedad opcional si `Type` se especifica como `AMAZON_COGNITO_USER_POOLS`.

Tipo: [UserPoolConfig](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [UserPoolConfig](#) propiedad de un `AWS::AppSync::GraphQLApi` recurso.

Ejemplos

Configura un tipo de autorización predeterminado y adicional

En este ejemplo, empezamos configurando un autorizador Lambda como el tipo de autorización predeterminado para nuestra API GraphQL.

```
AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31
...
Resources:
  MyGraphQLAPI:
    Type: AWS::Serverless::GraphQLApi
    Properties:
      Auth:
        Type: AWS_LAMBDA
        LambdaAuthorizer:
          AuthorizerUri: !GetAtt Authorizer1.Arn
          AuthorizerResultTtlInSeconds: 10
          IdentityValidationExpression: hello
```

A continuación, configuramos tipos de autorización adicionales para nuestra API GraphQL añadiendo lo siguiente a nuestra plantilla de AWS SAM :

```
Additional:
- Type: AWS_IAM
- Type: API_KEY
- Type: OPENID_CONNECT
  OpenIDConnect:
    AuthTTL: 10
    ClientId: myId
    IatTTL: 10
    Issuer: prod
```

Esto da como resultado la siguiente AWS SAM plantilla:

```
AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31
...
Resources:
  MyGraphQLAPI:
    Type: AWS::Serverless::GraphQLApi
    Properties:
      Auth:
        Type: AWS_LAMBDA
        LambdaAuthorizer:
          AuthorizerUri: !GetAtt Authorizer1.Arn
          AuthorizerResultTtlInSeconds: 10
          IdentityValidationExpression: hello
```

```
Additional:
- Type: AWS_IAM
- Type: API_KEY
- Type: OPENID_CONNECT
  OpenIDConnect:
    AuthTTL: 10
    ClientId: myId
    IatTTL: 10
    Issuer: prod
```

AuthProvider

Configuración de autorización opcional para sus tipos de autorización de GraphQL API adicionales.

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
LambdaAuthorizer: LambdaAuthorizerConfig
OpenIDConnect: OpenIDConnectConfig
Type: String
UserPool: UserPoolConfig
```

Propiedades

LambdaAuthorizer

Especifique la configuración de autorización opcional para su autorizador de funciones de AWS Lambda eventos. Puedes configurar esta propiedad opcional si `Type` se especifica como `AWS_LAMBDA`.

Tipo: [LambdaAuthorizerConfig](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [LambdaAuthorizerConfig](#) propiedad de un `AWS::AppSync::GraphQLApi` [AdditionalAuthenticationProvider](#) objeto.

OpenIDConnect

Especifica la configuración de autorización opcional para su servicio OpenID Connect compatible. Puedes configurar esta propiedad opcional si Type se especifica como OPENID_CONNECT.

Tipo: [OpenIDConnectConfig](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [OpenIDConnectConfig](#) propiedad de un `AWS::AppSync::GraphQLApi` [AdditionalAuthenticationProvider](#) objeto.

Type

El tipo de autorización predeterminado entre las aplicaciones y tu AWS AppSync GraphQL API.

Para obtener una lista y una descripción de los valores permitidos, consulta [Autorización y autenticación](#) en la Guía para desarrolladores de AWS AppSync .

Al especificar un autorizador de Lambda (AWS_LAMBDA), AWS SAM crea una AWS Identity and Access Management política (de IAM) para aprovisionar permisos entre la API GraphQL y la función de Lambda.

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [AuthenticationType](#) propiedad de un objeto. `AWS::AppSync::GraphQLApi` [AdditionalAuthenticationProvider](#)

UserPool

Especifica la configuración de autorización opcional para utilizar grupos de usuarios de Amazon Cognito. Puedes configurar esta propiedad opcional si Type se especifica como AMAZON_COGNITO_USER_POOLS.

Tipo: [UserPoolConfig](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [UserPoolConfig](#) propiedad de un `AWS::AppSync::GraphQLApi` [AdditionalAuthenticationProvider](#) objeto.

DataSource

Configura un origen de datos al que se pueda conectar su solucionador de API GraphQL. Puede usar las plantillas AWS Serverless Application Model (AWS SAM) para configurar las conexiones a las siguientes fuentes de datos:

- Amazon DynamoDB
- AWS Lambda

Para obtener más información sobre los orígenes de datos, consulta [Adjuntar un origen de datos](#) en la Guía para desarrolladores de AWS AppSync .

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
DynamoDb: DynamoDb  
Lambda: Lambda
```

Propiedades

DynamoDb

Configura una tabla de DynamoDB como origen de datos para su solucionador de API GraphQL.

Tipo: [DynamoDb](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

Lambda

Configure una función de eventos de Lambda como fuente de datos para su solucionador de GraphQL API.

Type: [Lambda](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

DynamoDb

Configura una tabla de Amazon DynamoDB como origen de datos para su solucionador de API de GraphQL.

Sintaxis

Para declarar esta entidad en su plantilla AWS Serverless Application Model (AWS SAM), utilice la siguiente sintaxis.

YAML

```
LogicalId:  
  DeltaSync: DeltaSyncConfig  
  Description: String  
  Name: String  
  Permissions: List  
  Region: String  
  ServiceRoleArn: String  
  TableArn: String  
  TableName: String  
  UseCallerCredentials: Boolean  
  Versioned: Boolean
```

Propiedades

DeltaSync

Describe una configuración de Delta Sync.

Tipo: [DeltaSyncConfig](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [DeltaSyncConfig](#) propiedad de un `AWS::AppSync::DataSource` DynamoDBConfig objeto.

Description

La descripción de su origen de datos.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Description](#) propiedad de un `AWS::AppSync::DataSource` recurso.

LogicalId

El nombre único de su origen de datos.

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Name](#) propiedad de un `AWS::AppSync::DataSource` recurso.

Name

El nombre de su origen de datos. Especifica esta propiedad para anular el valor LogicalId.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Name](#) propiedad de un `AWS::AppSync::DataSource` recurso.

Permissions

Aprovisione permisos a su origen de datos utilizando [Conectores AWS SAM](#). Puedes proporcionar cualquiera de los siguientes valores en una lista:

- Read – Permita que su solucionador lea su origen de datos.

- **Write** – Permita que su solucionador escriba su origen de datos.

AWS SAM utiliza un `AWS::Serverless::Connector` recurso que se transforma en el momento de la implementación para aprovisionar sus permisos. Para más información sobre los recursos generados, consulta [CloudFormation recursos generados cuando se especifica AWS::Serverless::Connector](#).

 Note

Puedes especificar `Permissions` o `ServiceRoleArn`, pero no ambos. Si no se especifica ninguno, AWS SAM generará los valores predeterminados de `Read` y `Write`. Para revocar el acceso a la fuente de datos, elimine el objeto `DynamoDB` de la plantilla. AWS SAM

Tipo: lista

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un equivalente. CloudFormation Es similar a la propiedad [Permissions](#) de un recurso `AWS::Serverless::Connector`.

Region

La Región de AWS de su tabla de DynamoDB. Si no lo especifica, AWS SAM usa. [AWS::Region](#)

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [AwsRegion](#) propiedad de un `AWS::AppSync::DataSource` `DynamoDBConfig` objeto.

ServiceRoleArn

El ARN del rol de servicio AWS Identity and Access Management (IAM) de la fuente de datos. El sistema adopta este rol al acceder al origen de datos.

Puedes especificar `Permissions` o `ServiceRoleArn`, pero no ambos.

Tipo: cadena

Requerido: No. Si no se especifica, AWS SAM aplica el valor predeterminado para `Permissions`.

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [ServiceRoleArn](#) propiedad de un `AWS::AppSync::DataSource` recurso.

TableArn

El ARN para la tabla de DynamoDB

Tipo: cadena

Obligatorio: condicional. Si no se especifica el `ServiceRoleArn`, la `TableArn` es obligatoria.

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

TableName

El nombre de la tabla.

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [TableName](#) propiedad de un `AWS::AppSync::DataSource` `DynamoDBConfig` objeto.

UseCallerCredentials

Establécelo en `true` para utilizar IAM con este origen de datos.

Tipo: Booleano

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [UseCallerCredentials](#) propiedad de un `AWS::AppSync::DataSource` `DynamoDBConfig` objeto.

Versioned

Establécelo en `true` para utilizar [Detección de conflictos, resolución de conflictos y sincronización](#) con este origen de datos.

Tipo: Booleano

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Versioned](#) propiedad de un `AWS::AppSync::DataSource` DynamoDBConfig objeto.

Lambda

Configura una función de AWS Lambda eventos como fuente de datos para tu solucionador de GraphQL API.

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
LogicalId:  
  Description: String  
  FunctionArn: String  
  Name: String  
  ServiceRoleArn: String
```

Propiedades

Description

La descripción de su origen de datos.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Description](#) propiedad de un `AWS::AppSync::DataSource` recurso.

FunctionArn

El ARN para la función de Lambda.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [LambdaFunctionArn](#) propiedad de un `AWS::AppSync::DataSource` `LambdaConfig` objeto.

LogicalId

El nombre único de su origen de datos.

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Name](#) propiedad de un `AWS::AppSync::DataSource` recurso.

Name

El nombre de su origen de datos. Especifica esta propiedad para anular el valor `LogicalId`.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Name](#) propiedad de un `AWS::AppSync::DataSource` recurso.

ServiceRoleArn

El ARN del rol de servicio AWS Identity and Access Management (IAM) de la fuente de datos. El sistema adopta este rol al acceder al origen de datos.

Note

Para revocar el acceso al origen de datos, elimine el objeto Lambda de la plantilla de AWS SAM .

Tipo: cadena

Requerido: No. Si no se especifica, AWS SAM proporcionará los Write permisos mediante [Conectores AWS SAM](#) .

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [ServiceRoleArn](#) propiedad de un `AWS::AppSync::DataSource` recurso.

Función

Configura funciones en las API de GraphQL para realizar ciertas operaciones.

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
LogicalId:  
  CodeUri: String  
  DataSource: String  
  Description: String  
  Id: String  
  InlineCode: String  
  MaxBatchSize: Integer  
  Name: String  
  Runtime: Runtime  
  Sync: SyncConfig
```

Propiedades

CodeUri

Una URL de Amazon Simple Storage Service (Amazon S3) es de código de función o es una ruta de acceso a una carpeta local.

Si especifica una ruta a una carpeta local, CloudFormation requiere que el archivo se cargue primero en Amazon S3 antes de la implementación. Puedes utilizar la AWS SAM CLI para facilitar este proceso. Para obtener más información, consulta [Cómo AWS SAM carga archivos locales en el momento de la implementación](#).

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [CodeS3Location](#) propiedad de un `AWS::AppSync::FunctionConfiguration` recurso.

DataSource

El nombre de origen de datos al que se asociará esta función.

- Para hacer referencia a un origen de datos dentro del recurso `AWS::Serverless::GraphQLApi`, especifique su ID lógico.
- Para hacer referencia a un origen de datos externo al recurso `AWS::Serverless::GraphQLApi`, proporcione su atributo `Name` mediante la función intrínseca `Fn::GetAtt`. Por ejemplo, `!GetAtt MyLambdaDataSource.Name`.
- Para hacer referencia a un origen de datos desde una pila diferente, utilice [Fn::ImportValue](#).

Si `[NONE | None | none]` se especifica una variación de, AWS SAM generará un `None` valor para el `AWS::AppSync::DataSource` [Type](#) objeto.

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [DataSourceName](#) propiedad de un `AWS::AppSync::FunctionConfiguration` recurso.

Description

La descripción de la función.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Description](#) propiedad de un `AWS::AppSync::FunctionConfiguration` recurso.

Id

El identificador de función de una función ubicada fuera del recurso `AWS::Serverless::GraphQLApi`.

- Para hacer referencia a una función dentro de la misma AWS SAM plantilla, utilice la función `Fn::GetAtt` intrínseca. Por ejemplo, `Id: !GetAtt createPostItemFunc.FunctionId`.
- Para hacer referencia a una función desde una pila diferente, utilice [Fn::ImportValue](#).

Cuando se usa `Id`, no se permiten todas las demás propiedades. AWS SAM pasará automáticamente el ID de función de la función a la que se hace referencia.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

InlineCode

El código de la función que contiene las funciones de solicitud y respuesta.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Code](#) propiedad de un `AWS::AppSync::FunctionConfiguration` recurso.

LogicalId

El nombre único de la función.

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Name](#) propiedad de un `AWS::AppSync::FunctionConfiguration` recurso.

MaxBatchSize

El número máximo de entradas de solicitudes de resolución que se enviarán a una sola función de AWS Lambda evento en una BatchInvoke operación.

Tipo: entero

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [MaxBatchSize](#) propiedad de un `AWS::AppSync::FunctionConfiguration` recurso.

Name

El nombre de la función. Especifica si desea anular el valor de LogicalId.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Name](#) propiedad de un `AWS::AppSync::FunctionConfiguration` recurso.

Runtime

Describe el tiempo de ejecución utilizado por una AWS AppSync función o un solucionador de AWS AppSync canalizaciones. Especifica el nombre y la versión del tiempo de ejecución que se utilizará.

Tipo: [tiempo de ejecución](#)

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente. Es similar a la propiedad [Runtime](#) de un recurso `AWS::AppSync::FunctionConfiguration`.

Sync

Describe una configuración de sincronización para una función.

Especifica la estrategia de detección de conflictos y la estrategia de resolución que se utilizará cuando se invoque la función.

Tipo: [SyncConfig](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [SyncConfig](#) propiedad de un `AWS::AppSync::FunctionConfiguration` recurso.

Tiempo de ejecución

El tiempo de ejecución de su función o solucionador de canalización. Especifica el nombre y la versión que se utilizará.

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
Name: String  
Version: String
```

Propiedades

Name

El nombre del tiempo de ejecución que se utilizará. El único valor permitido actualmente es APPSYNC_JS.

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Name](#) propiedad de un `AWS::AppSync::FunctionConfiguration` AppSyncRuntime objeto.

Version

La versión del tiempo de ejecución que se utilizará. En la actualidad, la única versión permitida es 1.0.0.

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [RuntimeVersion](#) propiedad de un `AWS::AppSync::FunctionConfiguration` AppSyncRuntime objeto.

Solucionador

Configura los resolvers para los campos de tu API. GraphQL AWS Serverless Application Model (AWS SAM) admite [solucionadores de JavaScript canalizaciones](#).

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
OperationType:  
  LogicalId:  
    Caching: CachingConfig  
    CodeUri: String  
    FieldName: String  
    InlineCode: String  
    MaxBatchSize: Integer  
    Pipeline: List  
    Runtime: Runtime  
    Sync: SyncConfig
```

Propiedades

Caching

La configuración de almacenamiento en caché para un solucionador que tiene la caché activada.

Tipo: [CachingConfig](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [CachingConfig](#) propiedad de un `AWS::AppSync::Resolver` recurso.

CodeUri

El código de la función de solucionador es el URI de Amazon Simple Storage Service (Amazon S3) o la ruta a una carpeta local.

Si especifica una ruta a una carpeta local, CloudFormation requiere que el archivo se cargue primero en Amazon S3 antes de la implementación. Puedes utilizar la AWS SAM CLI para facilitar este proceso. Para obtener más información, consulte [Cómo AWS SAM carga archivos locales en el momento de la implementación](#).

Si no `InlineCode` se proporciona `CodeUri` ninguna de las dos, AWS SAM generará un `InlineCode` generador que redireccionará la solicitud a la primera función de canalización y recibirá la respuesta de la última función de canalización.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [CodeS3Location](#) propiedad de un `AWS::AppSync::Resolver` recurso.

FieldName

El nombre de su solucionador. Especifica esta propiedad para anular el valor `LogicalId`.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [FieldName](#) propiedad de un `AWS::AppSync::Resolver` recurso.

InlineCode

El código de solucionador que contiene las funciones de solicitud y respuesta.

Si no `InlineCode` se proporciona `CodeUri` ninguna de ellas, se AWS SAM generará un `InlineCode` generador que redireccionará la solicitud a la primera función de canalización y recibirá la respuesta de la última función de canalización.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [Code](#) propiedad de un `AWS::AppSync::Resolver` recurso.

LogicalId

El nombre único para su solucionador. En un esquema GraphQL, el nombre del solucionador debe coincidir con el nombre del campo para el que se usa. Usa el mismo nombre de campo para `LogicalId`.

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

MaxBatchSize

El número máximo de entradas de solicitudes de resolución que se enviarán a una sola función de AWS Lambda evento en una BatchInvoke operación.

Tipo: entero

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [MaxBatchSize](#) propiedad de un `AWS::AppSync::Resolver` recurso.

OperationType

El tipo de operación GraphQL que está asociado al solucionador. Por ejemplo, Query, Mutation o Subscription. Puede agrupar varios solucionadores LogicalId en un solo OperationType.

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [TypeName](#) propiedad de un `AWS::AppSync::Resolver` recurso.

Pipeline

Funciones vinculadas con el solucionador de canalización. Especifica las funciones mediante un identificador lógico en una lista.

Tipo: lista

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente. Es similar a la propiedad [PipelineConfig](#) de un recurso `AWS::AppSync::Resolver`.

Runtime

El tiempo de ejecución de su función o solucionador de canalización. Especifica el nombre y la versión que se utilizará.

Tipo: [tiempo de ejecución](#)

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente. Es similar a la propiedad [Runtime](#) de un recurso `AWS::AppSync::Resolver`.

Sync

Describe una configuración de sincronización para un solucionador.

Especifica la estrategia de detección de conflictos y la estrategia de resolución que se utilizará cuando se invoque el resolver.

Tipo: [SyncConfig](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [SyncConfig](#) propiedad de un `AWS::AppSync::Resolver` recurso.

Ejemplos

Use la AWS SAM generó el código de la función de resolución y guardó los campos como variables

Este es el esquema GraphQL de nuestro ejemplo:

```
schema {
  query: Query
  mutation: Mutation
}

type Query {
  getPost(id: ID!): Post
}

type Mutation {
  addPost(author: String!, title: String!, content: String!): Post!
}

type Post {
  id: ID!
  author: String
  title: String
```



```
content: String
}
```

Aquí hay un fragmento de nuestra AWS SAM plantilla:

```
AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31
...
Resources:
  MyGraphQLApi:
    Type: AWS::Serverless::GraphQLApi
    Properties:
      ...
      Functions:
        preprocessPostItem:
          ...
        createPostItem:
          ...
      Resolvers:
        Mutation:
          addPost:
            Runtime:
              Name: APPSYNC_JS
              Version: 1.0.0
            Pipeline:
              - preprocessPostItem
              - createPostItem
```

En nuestra AWS SAM plantilla, no `CodeUri` especificamos o. `InlineCode` En el momento de la implementación, genera AWS SAM automáticamente el siguiente código en línea para nuestro solucionador:

```
export function request(ctx) {
  return {};
}

export function response(ctx) {
  return ctx.prev.result;
}
```

Este código de solucionador predeterminado redirige la solicitud a la primera función de canalización y recibe la respuesta de la última función de canalización.

En nuestra primera función de canalización, podemos usar el campo `args` proporcionado para analizar el objeto de solicitud y crear nuestras variables. Luego podemos usar estas variables dentro de nuestra función. A continuación, se muestra un ejemplo de nuestra función `preprocessPostItem`:

```
import { util } from "@aws-appsync/utils";

export function request(ctx) {
  const author = ctx.args.author;
  const title = ctx.args.title;
  const content = ctx.args.content;

  // Use variables to process data
}

export function response(ctx) {
  return ctx.result;
}
```

Tiempo de ejecución

El tiempo de ejecución de su función o solucionador de canalización. Especifica el nombre y la versión que se utilizará.

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
Name: String
Version: String
```

Propiedades

Name

El nombre del tiempo de ejecución que se utilizará. El único valor permitido actualmente es `APPSYNC_JS`.

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Name](#) propiedad de un `AWS::AppSync::Resolver` `AppSyncRuntime` objeto.

Version

La versión del tiempo de ejecución que se utilizará. En la actualidad, la única versión permitida es `1.0.0`.

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [RuntimeVersion](#) propiedad de un `AWS::AppSync::Resolver` `AppSyncRuntime` objeto.

AWS::Serverless::HttpApi

Cree una API HTTP de Amazon API Gateway, que permite crear API RESTful con menor latencia y costos que las API de REST. Para obtener más información, consulta [Trabajar con API HTTP](#) en la Guía del desarrollador de API Gateway.

Te recomendamos que utilices CloudFormation enlaces o políticas de IAM para comprobar que los recursos de API Gateway tienen autorizadores adjuntos para controlar el acceso a ellos.

Para obtener más información sobre el uso de CloudFormation ganchos, consulte [Registrar ganchos](#) en la guía del usuario de la CloudFormation CLI y en el repositorio [GitHub apigw-enforce-authorizer](#).

Para obtener más información sobre el uso de las políticas de IAM, consulta [Exigir que las rutas de API tengan autorización](#) en la Guía para desarrolladores de API Gateway.

Note

Cuando despliegas en, transforma tus recursos en AWS CloudFormation recursos AWS SAM . AWS SAM CloudFormation Para obtener más información, consulte [Generado CloudFormation recursos para AWS SAM](#).

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
Type: AWS::Serverless::HttpApi
Properties:
  AccessLogSettings: AccessLogSettings
  Auth: HttpApiAuth
  CorsConfiguration: String | HttpApiCorsConfiguration
  DefaultRouteSettings: RouteSettings
  DefinitionBody: JSON
  DefinitionUri: String | HttpApiDefinition
  Description: String
  DisableExecuteApiEndpoint: Boolean
  Domain: HttpApiDomainConfiguration
  FailOnWarnings: Boolean
  Name: String
  PropagateTags: Boolean
  RouteSettings: RouteSettings
  StageName: String
  StageVariables: Json
  Tags: Map
```

Propiedades

AccessLogSettings

Configuración del registro de acceso en una etapa.

Tipo: [AccessLogSettings](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [AccessLogSettings](#) propiedad de un `AWS::ApiGatewayV2::Stage` recurso.

Auth

Configure la autorización para controlar el acceso a la API HTTP de API Gateway.

Para obtener más información, consulta [Control del acceso a las API HTTP con autorizadores de JWT](#) en la Guía para desarrolladores de API Gateway.

Tipo: [HttpApiAuth](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

CorsConfiguration

Administre el uso compartido de recursos entre orígenes (CORS) para todas las API HTTP Gateway. Especifica el dominio que quieres permitir como cadena o especifica un objeto `HttpApiCorsConfiguration`. Tenga en cuenta que CORS requiere AWS SAM modificar su definición de OpenAPI, por lo que CORS solo funciona si se especifica `DefinitionBody` la propiedad.

Para obtener más información, consulta [Configuración de CORS para API HTTP](#) en la Guía para desarrolladores de API Gateway.

Note

Si `CorsConfiguration` se establece tanto en una definición de OpenAPI como en el nivel de propiedad, entonces AWS SAM fusiona ambas fuentes de configuración dando prioridad a las propiedades. Si esta propiedad está establecida en `true`, se permiten todos los orígenes.

Tipo: Cadena | [HttpApiCorsConfiguration](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

DefaultRouteSettings

Es la configuración de ruta predeterminada para esta API HTTP. Esta configuración se aplica a todas las rutas, a menos que la propiedad `RouteSettings` la anula para determinadas rutas.

Tipo: [RouteSettings](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [RouteSettings](#) propiedad de un `AWS::ApiGatewayV2::Stage` recurso.

DefinitionBody

La definición de OpenAPI que describe tu API HTTP. Si no especificas una `DefinitionUri` o una `DefinitionBody`, AWS SAM genera una `DefinitionBody` para ti en función de la configuración de la plantilla.

Tipo: JSON

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es similar a la [Body](#) propiedad de un `AWS::ApiGatewayV2::Api` recurso. Si se proporcionan determinadas propiedades, AWS SAM puede insertar contenido en ellas o modificarlas `DefinitionBody` antes de pasarlas a ellas CloudFormation. Las propiedades incluyen `Auth` un `EventSource` tipo `HttpApi` para el `AWS::Serverless::Function` recurso correspondiente.

DefinitionUri

El URI, ruta de archivo local u objeto de ubicación de Amazon Simple Storage Service (Amazon S3) de la definición de OpenAPI que define la API HTTP. El objeto Amazon S3 al que hace referencia esta propiedad debe ser un archivo de definición de OpenAPI válido. Si no especifica una `DefinitionUri` o `DefinitionBody` se especifica una, AWS SAM genera una `DefinitionBody` para usted en función de la configuración de la plantilla.

Si proporcionas una ruta de archivo local, la plantilla debe seguir el flujo de trabajo que incluye el comando `sam deploy` o `sam package` para que la definición se transforme correctamente.

Las funciones intrínsecas no se admiten en los archivos de OpenApi definición externos a los que haga referencia `DefinitionUri`. Para importar una OpenApi definición a la plantilla, utilice la `DefinitionBody` propiedad con la [transformación Include](#).

Tipo: Cadena | [HttpApiDefinition](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es similar a la [BodyS3Location](#) propiedad de un `AWS::ApiGatewayV2::Api` recurso. Las propiedades anidadas de Amazon S3 tienen un nombre diferente.

Description

La descripción del recurso de la API HTTP.

Cuando lo especifique `Description`, AWS SAM modificará la OpenApi definición del recurso de la API HTTP configurando el `description` campo. Las siguientes situaciones producirán un error:

- La `DefinitionBody` propiedad se especifica con el `description` campo establecido en la definición de API abierta. Esto provoca un conflicto en el `description` campo que AWS SAM no se resuelve.
- La `DefinitionUri` propiedad está especificada: AWS SAM no modificará una definición de API abierta que se recupere de Amazon S3.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

DisableExecuteApiEndpoint

Especifica si los clientes pueden invocar su API HTTP utilizando el `execute-api` punto de conexión `https://{api_id}.execute-api.{region}.amazonaws.com` predeterminado. De forma predeterminada, los clientes pueden invocar su API con el punto de enlace predeterminado. Para exigir que los clientes utilicen un nombre de dominio personalizado para invocar la API, desactive el punto de conexión predeterminado.

Para usar esta propiedad, debe especificar la propiedad `DefinitionBody` en lugar de la propiedad `DefinitionUri` o definir `x-amazon-apigateway-endpoint-configuration` con `disableExecuteApiEndpoint` en su definición de OpenAPI.

Tipo: Booleano

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es similar a la [DisableExecuteApiEndpoint](#) propiedad de un `AWS::ApiGatewayV2::Api` recurso. Se pasa directamente a la

propiedad `disableExecuteApiEndpoint` de una extensión [x-amazon-apigateway-endpoint-configuration](#), que se agrega a la propiedad `Body` de un recurso `AWS::ApiGatewayV2::Api`.

Domain

Configura un dominio personalizado para esta API HTTP de API Gateway.

Tipo: [HttpApiDomainConfiguration](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

FailOnWarnings

Especifica si se va a restaurar la creación de HTTP API (`true`) o no (`false`) cuando se encuentra una advertencia. El valor predeterminado es `false`.

Tipo: Booleano

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [FailOnWarnings](#) propiedad de un `AWS::ApiGatewayV2::Api` recurso.

Name

El nombre del recurso de HTTP API.

Cuando lo especifiques `Name`, AWS SAM modificará la definición de OpenAPI del recurso de la API HTTP configurando el `title` campo. Las siguientes situaciones producirán un error:

- La `DefinitionBody` propiedad se especifica con el `title` campo establecido en la definición de API abierta. Esto provoca un conflicto en el `title` campo que AWS SAM no se resuelve.
- La `DefinitionUri` propiedad está especificada: AWS SAM no modificará una definición de API abierta que se recupere de Amazon S3.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

PropagateTags

Indique si quiere pasar o no las etiquetas de la propiedad Tags a los recursos generados [AWS::Serverless::HttpApi](#). Especifica True para propagar las etiquetas en los recursos generados.

Tipo: Booleano

Obligatorio: no

Valor predeterminado: False

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

RouteSettings

La configuración de ruta, por ruta, para esta API HTTP. Para obtener más información, consulta [Trabajar con rutas para API HTTP](#) en la Guía para desarrolladores de API Gateway.

Tipo: [RouteSettings](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [RouteSettings](#) propiedad de un `AWS::ApiGatewayV2::Stage` recurso.

StageName

Es el nombre de la fase de la API. Si no se especifica ningún nombre, AWS SAM utiliza la `$default` etapa de API Gateway.

Tipo: cadena

Obligatorio: no

Valor predeterminado: `$default`

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [StageName](#) propiedad de un `AWS::ApiGatewayV2::Stage` recurso.

StageVariables

Una asignación que define las variables de la fase. Los nombres de las variables pueden tener caracteres alfanuméricos y de subrayado. Los valores deben coincidir con [A-Za-z0-9-._~:/? #&=, +].

Type: [Json](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [StageVariables](#) propiedad de un `AWS::ApiGatewayV2::Stage` recurso.

Tags

Un mapa (cadena a cadena) que especifica las etiquetas a añadir a esta etapa de API Gateway. Las claves pueden tener una longitud de entre 1 y 128 caracteres Unicode y no pueden incluir el prefijo `aws:`. Puede utilizar cualquiera de los siguientes caracteres: el conjunto de letras Unicode, dígitos, espacio en blanco, `_`, `.`, `/`, `=`, `+` y `-`. Los valores pueden tener de 1 a 256 caracteres Unicode.

Tipo: mapa

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

Notas adicionales: La `Tags` propiedad requiere AWS SAM modificar la definición de OpenAPI, por lo que las etiquetas se añaden solo si se especifica la `DefinitionBody` propiedad; no se añaden etiquetas si se especifica la propiedad `DefinitionUri`. AWS SAM añade automáticamente una etiqueta `httpapi:createdBy:SAM`. Las etiquetas también se añaden al recurso `AWS::ApiGatewayV2::Stage` y al recurso `AWS::ApiGatewayV2::DomainName` (si `DomainName` se especifica).

valores devueltos

Ref.

Cuando se pasa el ID lógico de este recurso a la función intrínseca `Ref`, `Ref` devuelve el ID de API del recurso subyacente `AWS::ApiGatewayV2::Api`, por ejemplo, `a1bcdef2gh`.

Para obtener más información sobre el uso de la función Ref, consulta [Ref](#) en la Guía del usuario de AWS CloudFormation .

Ejemplos

Sencillo HttpApi

El siguiente ejemplo muestra el mínimo necesario para configurar un punto final de la API HTTP respaldado por una función de eventos de Lambda. En este ejemplo, se utiliza la API HTTP predeterminada que AWS SAM crea.

YAML

```
AWSTemplateFormatVersion: '2010-09-09'
Description: AWS SAM template with a simple API definition
Resources:
  ApiFunction:
    Type: AWS::Serverless::Function
    Properties:
      Events:
        ApiEvent:
          Type: HttpApi
      Handler: index.handler
      InlineCode: |
        def handler(event, context):
          return {'body': 'Hello World!', 'statusCode': 200}
      Runtime: python3.7
    Transform: AWS::Serverless-2016-10-31
```

HttpApi con Auth

El siguiente ejemplo muestra cómo configurar la autorización en puntos de conexión de API HTTP.

YAML

```
Properties:
  FailOnWarnings: true
  Auth:
    DefaultAuthorizer: OAuth2
    Authorizers:
      OAuth2:
```

```
AuthorizationScopes:
  - scope4
JwtConfiguration:
  issuer: "https://www.example.com/v1/connect/oauth2"
  audience:
    - MyApi
IdentitySource: "$request.querystring.param"
```

HttpApi con definición de OpenAPI

El siguiente ejemplo muestra cómo añadir una definición de OpenAPI a la plantilla.

Tenga en cuenta que AWS SAM rellena las integraciones de Lambda que falten para HttpApi los eventos que hacen referencia a esta API HTTP. AWS SAM también agrega las rutas faltantes a las que hacen referencia HttpApi los eventos.

YAML

```
Properties:
  FailOnWarnings: true
DefinitionBody:
  info:
    version: '1.0'
    title:
      Ref: AWS::StackName
  paths:
    "/":
      get:
        security:
          - OpenIdAuth:
              - scope1
              - scope2
        responses: {}
  openapi: 3.0.1
  securitySchemes:
    OpenIdAuth:
      type: openIdConnect
      x-amazon-apigateway-authorizer:
        identitySource: "$request.querystring.param"
        type: jwt
        jwtConfiguration:
          audience:
            - MyApi
```

```

    issuer: https://www.example.com/v1/connect/oidc
    openIdConnectUrl: https://www.example.com/v1/connect/oidc/.well-known/openid-configuration

```

HttpApi con ajustes de configuración

El siguiente ejemplo muestra cómo añadir la API HTTP y configuraciones de escenario a la plantilla.

YAML

```

AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31
Parameters:
  StageName:
    Type: String
    Default: Prod

Resources:
  HttpApiFunction:
    Type: AWS::Serverless::Function
    Properties:
      InlineCode: |
        def handler(event, context):
            import json
            return {
                "statusCode": 200,
                "body": json.dumps(event),
            }
      Handler: index.handler
      Runtime: python3.7
    Events:
      ExplicitApi: # warning: creates a public endpoint
        Type: HttpApi
        Properties:
          ApiId: !Ref HttpApi
          Method: GET
          Path: /path
          TimeoutInMillis: 15000
          PayloadFormatVersion: "2.0"
          RouteSettings:
            ThrottlingBurstLimit: 600

  HttpApi:

```

```

Type: AWS::Serverless::HttpApi
Properties:
  StageName: !Ref StageName
  Tags:
    Tag: Value
  AccessLogSettings:
    DestinationArn: !GetAtt AccessLogs.Arn
    Format: $context.requestId
  DefaultRouteSettings:
    ThrottlingBurstLimit: 200
  RouteSettings:
    "GET /path":
      ThrottlingBurstLimit: 500 # overridden in HttpApi Event
  StageVariables:
    StageVar: Value
  FailOnWarnings: true

AccessLogs:
  Type: AWS::Logs::LogGroup

Outputs:
  HttpApiUrl:
    Description: URL of your API endpoint
    Value:
      Fn::Sub: 'https://${HttpApi}.execute-api.${AWS::Region}.${AWS::URLSuffix}/${StageName}/'
  HttpApiId:
    Description: Api id of HttpApi
    Value:
      Ref: HttpApi

```

HttpApiAuth

Configure la autorización para controlar el acceso a la API HTTP de Amazon API Gateway.

Para obtener más información sobre la configuración del acceso a HTTP APIs, consulte [Control y administración del acceso a una API HTTP en API Gateway](#) en la Guía para desarrolladores de API Gateway.

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
Authorizers: OAuth2Authorizer | LambdaAuthorizer  
DefaultAuthorizer: String  
EnableIamAuthorizer: Boolean
```

Propiedades

Authorizers

El autorizador utilizado para controlar el acceso a la API de API Gateway.

Tipo: [OAuth2Autorizador](#) | [LambdaAuthorizer](#)

Obligatorio: no

Valor predeterminado: Ninguno

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

Notas adicionales: AWS SAM añade los autorizadores a la definición de OpenAPI.

DefaultAuthorizer

Especifica el autorizador predeterminado que se utilizará para autorizar las llamadas a la API de API Gateway. Puede especificar `AWS_IAM` como autorizador predeterminado si `EnableIamAuthorizer` está configurado en `true`. De lo contrario, especifica un autorizador que haya definido en `Authorizers`.

Tipo: cadena

Obligatorio: no

Valor predeterminado: Ninguno

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

EnableIamAuthorizer

Especifica si quieres utilizar la autorización de IAM para la ruta de la API.

Tipo: Booleano

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

Ejemplos

OAuth 2.0: Autorizador

OAuth Ejemplo de autorizador 2.0

YAML

```
Auth:
  Authorizers:
    OAuth2Authorizer:
      AuthorizationScopes:
        - scope1
        - scope2
      JwtConfiguration:
        issuer: "https://www.example.com/v1/connect/oauth2"
        audience:
          - MyApi
      IdentitySource: "$request.querystring.param"
  DefaultAuthorizer: OAuth2Authorizer
```

Autorizador de IAM

Ejemplo de autorizador de IAM

YAML

```
Auth:
  EnableIamAuthorizer: true
  DefaultAuthorizer: AWS_IAM
```

LambdaAuthorizer

Configure un autorizador Lambda para controlar el acceso a la API HTTP de Amazon API Gateway con una AWS Lambda función de eventos.

Para obtener más información y ejemplos, consulte Cómo [trabajar con AWS Lambda autorizadores para las API HTTP](#) en la Guía para desarrolladores de API Gateway.

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
AuthorizerPayloadFormatVersion: String  
EnableFunctionDefaultPermissions: Boolean  
EnableSimpleResponses: Boolean  
FunctionArn: String  
FunctionInvokeRole: String  
Identity: LambdaAuthorizationIdentity
```

Propiedades

AuthorizerPayloadFormatVersion

Especifica el formato de la carga enviada a un autorizador de Lambda API HTTP. Necesario para los autorizadores de Lambda API HTTP.

Esto se transfiere a la sección `authorizerPayloadFormatVersion` de un `x-amazon-apigateway-authorizer` en la sección `securitySchemes` de la definición de OpenAPI.

Valores válidos: `1.0` o `2.0`

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

EnableFunctionDefaultPermissions

De forma predeterminada, el recurso de la API HTTP no tiene permiso para invocar el autorizador Lambda. Especifica esta propiedad como `true` para crear automáticamente permisos entre su recurso de API HTTP y su autorizador Lambda.

Tipo: Booleano

Obligatorio: no

Valor predeterminado: `false`

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

`EnableSimpleResponses`

Especifica si un autorizador de Lambda devuelve una respuesta en un formato simple. De forma predeterminada, un autorizador Lambda debe devolver una política AWS Identity and Access Management (IAM). Si está habilitado, el autorizador de Lambda puede devolver un valor booleano en lugar de una política de IAM.

Esto se transfiere a la sección `enableSimpleResponses` de un `x-amazon-apigateway-authorizer` en la sección `securitySchemes` de la definición de OpenAPI.

Tipo: Booleano

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un equivalente. CloudFormation

`FunctionArn`

El nombre de recurso de Amazon (ARN) de la función de Lambda que proporciona autorización para la API.

Esto se transfiere a la sección `authorizerUri` de un `x-amazon-apigateway-authorizer` en la sección `securitySchemes` de la definición de OpenAPI.

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

`FunctionInvokeRole`

El ARN del rol de IAM que tiene las credenciales necesarias para API Gateway para invocar a la función autorizadora. Especifica este parámetro si la política basada en recursos de una función no concede el permiso `lambda:InvokeFunction` a API Gateway.

Esto se transfiere a la sección `authorizerCredentials` de un `x-amazon-apigateway-authorizer` en la sección `securitySchemes` de la definición de OpenAPI.

Para obtener más información, consulta [Creación de un autorizador Lambda](#) en la Guía para desarrolladores de API Gateway.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

Identity

Especifica un autorizador `IdentitySource` en una solicitud entrante.

Esto se transfiere a la sección `identitySource` de un `x-amazon-apigateway-authorizer` en la sección `securitySchemes` de la definición de OpenAPI.

Tipo: [LambdaAuthorizationIdentity](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

Ejemplos

LambdaAuthorizer

LambdaAuthorizer ejemplo

YAML

```
Auth:
  Authorizers:
    MyLambdaAuthorizer:
      AuthorizerPayloadFormatVersion: 2.0
      FunctionArn:
        Fn::GetAtt:
          - MyAuthFunction
```

```
- Arn
FunctionInvokeRole:
  Fn::GetAtt:
    - LambdaAuthInvokeRole
    - Arn
Identity:
  Headers:
    - Authorization
```

LambdaAuthorizationIdentity

La propiedad Use se puede usar para especificar una IdentitySource en una solicitud entrante de un autorizador Lambda. Para obtener más información sobre fuentes de identidad, consulta [Fuentes de identidad](#) en la Guía del desarrollador de API Gateway.

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
Context: List
Headers: List
QueryStrings: List
ReauthorizeEvery: Integer
StageVariables: List
```

Propiedades

Context

Convierte las cadenas de contexto dadas en una lista de expresiones de mapeo en el formato `$context.contextString`.

Tipo: lista

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

Headers

Convierte los encabezados en una lista de expresiones de mapeo en el formato `$request.header.name`.

Tipo: Lista

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

QueryString

Convierte las cadenas de consulta dadas en una lista de expresiones de mapeo en el formato `$request.querystring.queryString`.

Tipo: Lista

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

ReauthorizeEvery

El período time-to-live (TTL), en segundos, que especifica cuánto tiempo API Gateway almacena en caché los resultados del autorizador. Si especificas un valor mayor que 0, API Gateway almacena en caché las respuestas del autorizador. El valor máximo es 3600 o 1 hora.

Tipo: entero

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un equivalente. CloudFormation

StageVariables

Convierte las variables de etapa dadas en una lista de expresiones de mapeo en el formato `$stageVariables.stageVariable`.

Tipo: Lista

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

Ejemplos

LambdaRequestIdentity

Ejemplo de identidad de solicitud de Lambda

YAML

```
Identity:
  QueryStrings:
    - auth
  Headers:
    - Authorization
  StageVariables:
    - VARIABLE
  Context:
    - authcontext
  ReauthorizeEvery: 100
```

OAuth2Authorizer

Definición de un autorizador OAuth 2.0, también conocido como autorizador de JSON Web Token (JWT).

Para obtener más información, consulte [Control del acceso a HTTP APIs con autorizadores JWT](#) en la Guía para desarrolladores de API Gateway.

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
AuthorizationScopes: List
IdentitySource: String
JwtConfiguration: Map
```

Propiedades

AuthorizationScopes

Lista de los ámbitos de autorización de este autorizador.

Tipo: lista

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

IdentitySource

Expresión de origen de identidad para este autorizador.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

JwtConfiguration

Configuración JWT para este autorizador.

Esto se transfiere a la sección `jwtConfiguration` de un `x-amazon-apigateway-authorizer` en la sección `securitySchemes` de la definición de OpenAPI.

Note

Las propiedades `issuer` y `audience` no distinguen entre mayúsculas y minúsculas y se pueden usar en minúsculas como en OpenAPI o en mayúsculas y como en. `Issuer Audience` [AWS::ApiGatewayV2::Authorizer](#)

Tipo: mapa

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un equivalente. CloudFormation

Ejemplos

OAuth Autorizador 2.0

OAuth Ejemplo de autorizador 2.0

YAML

```
Auth:
  Authorizers:
    OAuth2Authorizer:
      AuthorizationScopes:
        - scope1
      JwtConfiguration:
        issuer: "https://www.example.com/v1/connect/oauth2"
        audience:
          - MyApi
      IdentitySource: "$request.querystring.param"
  DefaultAuthorizer: OAuth2Authorizer
```

HttpApiCorsConfiguration

Administre el intercambio de recursos entre orígenes (CORS) para su HTTP. APIs Especifica el dominio que quieres permitir como cadena o especifica un diccionario con una configuración de Cors adicional. NOTA: Cors requiere que SAM modifique la definición de OpenAPI, por lo que solo funciona con los elementos OpenApi en línea definidos en la propiedad. `DefinitionBody`

Para obtener más información acerca de las API de HTTP, consulta [API de HTTP](#) en la Guía para desarrolladores de API Gateway.

Nota: Si `HttpApiCorsConfiguration` se establece tanto en OpenAPI como en el nivel de propiedad, las AWS SAM fusiona con las propiedades que tienen prioridad.

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
AllowCredentials: Boolean
AllowHeaders: List
AllowMethods: List
```


AllowOrigins: *List*
ExposeHeaders: *List*
MaxAge: *Integer*

Propiedades

AllowCredentials

Especifica si las credenciales están incluidas en la solicitud de CORS.

Tipo: Booleano

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

AllowHeaders

Representa una colección de encabezados permitidos.

Tipo: lista

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

AllowMethods

Representa una colección de métodos HTTP permitidos.

Tipo: lista

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

AllowOrigins

Representa una colección de orígenes permitidos.

Tipo: lista

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

ExposeHeaders

Representa una colección de encabezados expuestos.

Tipo: lista

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

MaxAge

Número de segundos durante los que el navegador debe almacenar en caché los resultados de la solicitud preliminar.

Tipo: entero

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

Ejemplos

HttpApiCorsConfiguration

Ejemplo de configuración básica de la API HTTP.

YAML

```
CorsConfiguration:
  AllowOrigins:
    - "https://example.com"
  AllowHeaders:
    - x-apigateway-header
  AllowMethods:
    - GET
  MaxAge: 600
  AllowCredentials: true
```

HttpApiDefinition

Un documento de OpenAPI que define la API.

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
Bucket: String  
Key: String  
Version: String
```

Propiedades

Bucket

Nombre del bucket de Amazon S3 donde se almacena el archivo OpenAPI.

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Bucket](#) propiedad del tipo de `AWS::ApiGatewayV2::Api` `BodyS3Location` datos.

Key

La clave de Amazon S3 del archivo de OpenAPI.

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Key](#) propiedad del tipo de `AWS::ApiGatewayV2::Api` `BodyS3Location` datos.

Version

Para objetos con control de versiones, la versión del archivo de OpenAPI.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Version](#) propiedad del tipo de `AWS::ApiGatewayV2::Api` BodyS3Location datos.

Ejemplos

Ejemplo de definición de Uri

Ejemplo de definición de API

YAML

```
DefinitionUri:
  Bucket: sam-s3-demo-bucket-name
  Key: mykey-name
  Version: 121212
```

HttpApiDomainConfiguration

Configura un dominio personalizado para una API.

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
BasePath: List
CertificateArn: String
DomainName: String
EndpointConfiguration: String
MutualTlsAuthentication: MutualTlsAuthentication
OwnershipVerificationCertificateArn: String
Route53: Route53Configuration
SecurityPolicy: String
```

Propiedades

BasePath

Una lista de las rutas base que se deben configurar con el nombre de dominio de Amazon API Gateway.

Tipo: lista

Obligatorio: no

Valor predeterminado: /

CloudFormation compatibilidad: esta propiedad es similar a la [ApiMappingKey](#) propiedad de un `AWS::ApiGatewayV2::ApiMapping` recurso. AWS SAM crea varios `AWS::ApiGatewayV2::ApiMapping` recursos, uno por cada valor especificado en esta propiedad.

CertificateArn

El nombre de recurso de Amazon (ARN) de un certificado AWS administrado para el punto final de este nombre de dominio. AWS Gestor de certificados es la única fuente compatible.

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [CertificateArn](#) propiedad de un `AWS::ApiGatewayV2::DomainName` `DomainNameConfiguration` recurso.

DomainName

El nombre de dominio personalizado para su API de API Gateway. No se admiten letras mayúsculas.

AWS SAM genera un `AWS::ApiGatewayV2::DomainName` recurso cuando se establece esta propiedad. Para obtener información acerca de esta situación, consulta [DomainName se especifica la propiedad](#). Para obtener información sobre CloudFormation los recursos generados, consulte [Generado CloudFormation recursos para AWS SAM](#).

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [DomainName](#) propiedad de un `AWS::ApiGatewayV2::DomainName` recurso.

EndpointConfiguration

Define el tipo de punto de conexión de API Gateway que se va a asignar al dominio personalizado. El valor de esta propiedad determina cómo se mapea la `CertificateArn` propiedad. CloudFormation

El único valor válido para HTTP APIs es `REGIONAL`.

Tipo: cadena

Obligatorio: no

Valor predeterminado: `REGIONAL`

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

`MutualTlsAuthentication`

La configuración de autenticación mutua de seguridad de la capa de transporte (TLS) para un nombre de dominio personalizado.

Tipo: [`MutualTlsAuthentication`](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [`MutualTlsAuthentication`](#) propiedad de un `AWS::ApiGatewayV2::DomainName` recurso.

`OwnershipVerificationCertificateArn`

El ARN del certificado público emitido por ACM para validar la propiedad de su dominio personalizado. Solo se requiere cuando se configura TLS mutua y se especifica un ARN de certificado de CA privado o importado de ACM para `CertificateArn`.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [`OwnershipVerificationCertificateArn`](#) propiedad del tipo de `AWS::ApiGatewayV2::DomainName` `DomainNameConfiguration` datos.

`Route53`

Define una configuración de Amazon Route 53.

Tipo: [`Route53Configuration`](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

SecurityPolicy

La versión TLS de la política de seguridad para este nombre de dominio.

El único valor válido para HTTP APIs es TLS_1_2.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [SecurityPolicy](#) propiedad del tipo de AWS::ApiGatewayV2::DomainName DomainNameConfiguration datos.

Ejemplos

DomainName

DomainName ejemplo

YAML

```
Domain:
  DomainName: www.example.com
  CertificateArn: arn-example
  EndpointConfiguration: REGIONAL
  Route53:
    HostedZoneId: Z1PA6795UKMFR9
  BasePath:
    - foo
    - bar
```

Route53Configuration

Configura los conjuntos de registros de Route53 para una API.

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
DistributionDomainName: String  
EvaluateTargetHealth: Boolean  
HostedZoneId: String  
HostedZoneName: String  
IpV6: Boolean  
Region: String  
SetIdentifier: String
```

Propiedades

DistributionDomainName

Configura una distribución personalizada del nombre de dominio personalizado de la API.

Tipo: cadena

Obligatorio: no

Predeterminado: utilice la distribución de API Gateway.

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [DNSName](#) propiedad de un `AWS::Route53::RecordSetGroup` `AliasTarget` recurso.

Notas adicionales: El nombre de dominio de una [CloudFront distribución](#).

EvaluateTargetHealth

Si `EvaluateTargetHealth` es verdadero, un registro de alias hereda el estado del AWS recurso al que se hace referencia, como un balanceador de cargas de Elastic Load Balancing u otro registro de la zona alojada.

Tipo: Booleano

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [EvaluateTargetHealth](#) propiedad de un `AWS::Route53::RecordSetGroup` `AliasTarget` recurso.

Notas adicionales: No se puede `EvaluateTargetHealth` establecer en `true` cuando el alias objetivo es una CloudFront distribución.

HostedZoneId

ID de la zona alojada en la que desea crear registros.

Especifica HostedZoneName o HostedZoneId, pero no ambos. Si tiene múltiples zonas hospedadas con el mismo nombre de dominio, debe especificar la zona hospedada mediante HostedZoneId.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [HostedZoneId](#) propiedad de un `AWS::Route53::RecordSetGroup` `RecordSet` recurso.

HostedZoneName

El nombre de la zona alojada en la que desea crear registros. Debe incluir un punto final (por ejemplo, `www.example.com.`) como parte del HostedZoneName.

Especifica HostedZoneName o HostedZoneId, pero no ambos. Si tiene múltiples zonas hospedadas con el mismo nombre de dominio, debe especificar la zona hospedada mediante HostedZoneId.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [HostedZoneName](#) propiedad de un `AWS::Route53::RecordSetGroup` `RecordSet` recurso.

IPv6

Cuando se establece esta propiedad, AWS SAM crea un `AWS::Route53::RecordSet` recurso y establece [Type](#) en AAAA el valor proporcionado HostedZone.

Tipo: Booleano

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

Region

Solo conjuntos de registros de recursos basados en la latencia: la EC2 región de Amazon en la que creó el recurso al que hace referencia este conjunto de registros de recursos. El recurso suele ser un AWS recurso, como una EC2 instancia o un balanceador de cargas ELB, y se hace referencia a él mediante una dirección IP o un nombre de dominio DNS, según el tipo de registro.

Cuando Amazon Route 53 recibe una consulta de DNS para un nombre y un tipo de dominio para los que ha creado conjuntos de registros de recursos de latencia, Route 53 selecciona el conjunto de registros de recursos de latencia que tiene la latencia más baja entre el usuario final y la EC2 región de Amazon asociada. Route 53 devuelve a continuación el valor asociado con el conjunto de registros de recursos seleccionados.

Tenga en cuenta lo siguiente:

- Solo puede especificar un `ResourceRecord` por conjunto de registros de recursos de latencia.
- Solo puedes crear un conjunto de registros de recursos de latencia para cada EC2 región de Amazon.
- No es necesario que crees conjuntos de registros de recursos de latencia para todas las EC2 regiones de Amazon. Route 53 elige la región con la mejor latencia entre las regiones para las que crea conjuntos de registros de recursos de latencia.
- No puede crear conjuntos de registros de recursos sin latencia que tengan los mismos valores `Name` y `Type` que los conjuntos de registros de recursos de latencia.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [Region](#) propiedad de un tipo de `AWS::Route53::RecordSetGroup` `RecordSet` datos.

SetIdentifier

Conjuntos de registros de recursos que tienen una política de direccionamiento que no es simple: Un identificador que diferencia entre varios conjuntos de registros de recursos con la misma combinación de nombre y tipo, como varios conjuntos de registros de recursos ponderados denominados `acme.example.com` de tipo `A`. En un grupo de conjuntos de registros de recursos que tengan el mismo nombre y tipo, el valor de `SetIdentifier` debe ser exclusivo para cada conjunto de registros de recursos.

Para obtener información sobre cómo dirigir las políticas, consulta [Elección de una política de direccionamiento](#) en la Guía para desarrolladores de Amazon Route 53.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [SetIdentifier](#) propiedad de un tipo de `AWS::Route53::RecordSetGroup` `RecordSet` datos.

Ejemplos

Ejemplo de configuración de Route 53

En este ejemplo se muestra cómo configurar Route 53.

YAML

```
Domain:
  DomainName: www.example.com
  CertificateArn: arn-example
  EndpointConfiguration: EDGE
  Route53:
    HostedZoneId: Z1PA6795UKMFR9
    EvaluateTargetHealth: true
    DistributionDomainName: xyz
```

AWS::Serverless::WebSocketApi

Crea una API de Amazon WebSocket API Gateway, que le permite crear aplicaciones de comunicación interactiva bidireccional. WebSocket Las API permiten al servidor enviar mensajes a los clientes sin que el cliente tenga que solicitarlos. Para obtener más información, consulte [Trabajar con WebSocket las API](#) en la Guía para desarrolladores de API Gateway.

Te recomendamos que utilices CloudFormation enlaces o políticas de IAM para comprobar que los recursos de API Gateway tienen autorizadores adjuntos para controlar el acceso a ellos.

Para obtener más información sobre el uso de CloudFormation ganchos, consulte [Registrar ganchos](#) en la guía del usuario de la CloudFormation CLI y en el repositorio [GitHub apigw-enforce-authorizer](#).

Para obtener más información sobre el uso de las políticas de IAM, consulta [Exigir que las rutas de API tengan autorización](#) en la Guía para desarrolladores de API Gateway.

 Note

Cuando despliegas en, transforma tus recursos en AWS CloudFormation recursos AWS SAM . AWS SAM CloudFormation Para obtener más información, consulte [Generado CloudFormation recursos para AWS SAM](#).

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
Type: AWS::Serverless::WebSocketApi
Properties:
  ApiKeySelectionExpression: String
  AccessLogSettings: AccessLogSettings
  Auth: WebSocketApiAuth
  DefaultRouteSettings: RouteSettings
  Description: String
  DisableExecuteApiEndpoint: Boolean
  DisableSchemaValidation: Boolean
  Domain: WebSocketApiDomainConfiguration
  IpAddressType: String
  Name: String
  PropagateTags: Boolean
  Routes: RouteConfiguration
  RouteSelectionExpression: String
  RouteSettings: RouteSettings
  StageName: String
  StageVariables: Json
  Tags: Map
```

Propiedades

ApiKeySelectionExpression

Es una expresión de selección de clave de API. Para obtener más información, consulte [Expresiones de selección de claves de API](#) en la Guía para desarrolladores de API Gateway.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [ApiKeySelectionExpression](#) propiedad de un `AWS::ApiGatewayV2::Api` recurso.

AccessLogSettings

Configuración del registro de acceso en una etapa.

Tipo: [AccessLogSettings](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [AccessLogSettings](#) propiedad de un `AWS::ApiGatewayV2::Stage` recurso.

Auth

Configura la autorización para controlar el acceso a tu WebSocket API. La autorización se aplica a la `$connect` ruta.

Para obtener más información, consulte [Control del acceso a WebSocket las API](#) en la Guía para desarrolladores de API Gateway.

Tipo: [WebSocketApiAuth](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

DefaultRouteSettings

La configuración de ruta predeterminada para esta WebSocket API. Esta configuración se aplica a todas las rutas, a menos que la propiedad `RouteSettings` la anula para determinadas rutas.

Tipo: [RouteSettings](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [DefaultRouteSettings](#) propiedad de un `AWS::ApiGatewayV2::Stage` recurso.

Description

Una descripción de la WebSocket API.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Description](#) propiedad de un `AWS::ApiGatewayV2::Api` recurso.

DisableExecuteApiEndpoint

Especifica si los clientes pueden invocar la API mediante el punto de conexión `execute-api` predeterminado. Para exigir que los clientes utilicen un nombre de dominio personalizado para invocar la API, desactive el punto de enlace predeterminado.

Tipo: Booleano

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [DisableExecuteApiEndpoint](#) propiedad de un `AWS::ApiGatewayV2::Api` recurso.

DisableSchemaValidation

Evite validar modelos al crear una implementación.

Tipo: Booleano

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [DisableSchemaValidation](#) propiedad de un `AWS::ApiGatewayV2::Api` recurso.

Domain

Configura un dominio personalizado para esta WebSocket API.

Note

WebSocket Las API no admiten la autenticación TLS mutua (MTLS). Si lo especifica `MutualTlsAuthentication` o `OwnershipVerificationCertificateArn`, AWS SAM devolverá un error.

Tipo: [WebSocketApiDomainConfiguration](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

IpAddressType

El tipo de dirección IP de la API. Los valores válidos son solo `ipv4` para IPv4 y `dualstack` para IPv4 e IPv6.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [IpAddressType](#) propiedad de un recurso. `AWS::ApiGatewayV2::Api`

Name

Un nombre para la WebSocket API. Si no especificas un nombre, AWS SAM generará uno para ti.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Name](#) propiedad de un `AWS::ApiGatewayV2::Api` recurso.

PropagateTags

Si `true`, AWS SAM agrega la `Tags` propiedad a los `AWS::ApiGatewayV2::DomainName` recursos `AWS::ApiGatewayV2::Stage` y que AWS SAM genera.

Tipo: Booleano

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

Routes

Las configuraciones de ruta para esta WebSocket API. Las rutas definen cómo se enrutan los mensajes a las funciones de eventos Lambda. Cada ruta consta de una clave de ruta y un ARN de función Lambda.

WebSocket Las API admiten tres rutas predefinidas: `$connect$disconnect`, y `$default`. También puede definir rutas personalizadas.

Tipo: [RouteConfiguration](#)

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

RouteSelectionExpression

La expresión de selección de rutas de la WebSocket API. Para obtener más información, consulte [Expresiones de selección de rutas](#) en la Guía para desarrolladores de API Gateway.

Un valor común es `$request.body.action` que enruta los mensajes en función de un `action` campo del cuerpo del mensaje.

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [RouteSelectionExpression](#) propiedad de un `AWS::ApiGatewayV2::Api` recurso.

RouteSettings

La configuración de ruta de esta WebSocket API. Esta configuración anula la `DefaultRouteSettings` de rutas específicas.

Tipo: [RouteSettings](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [RouteSettings](#) propiedad de un `AWS::ApiGatewayV2::Stage` recurso.

StageName

Es el nombre de la fase de la API. Si no especificas un nombre, se AWS SAM usa `default` como nombre artístico.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [StageName](#) propiedad de un `AWS::ApiGatewayV2::Stage` recurso.

StageVariables

Una asignación que define las variables de la fase. Los nombres de las variables pueden tener caracteres alfanuméricos y de subrayado, y los valores deben coincidir. [A-Za-z0-9-._~:/?#&=,]+

Type: [Json](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [StageVariables](#) propiedad de un `AWS::ApiGatewayV2::Stage` recurso.

Tags

Un mapa (cadena a cadena) que especifica las etiquetas que se van a añadir a esta WebSocket API. Para obtener más información sobre las claves y los valores válidos de las etiquetas, consulta la [etiqueta de recursos](#) en la Guía del usuario de CloudFormation .

Tipo: mapa

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

valores devueltos

Ref.

Cuando se pasa el ID lógico de este recurso a la función intrínseca `Ref`, `Ref` devuelve el ID de API del recurso subyacente `AWS::ApiGatewayV2::Api`, por ejemplo, `a1bcdef2gh`.

Para obtener más información sobre el uso de la función `Ref`, consulta [Ref](#) en la Guía del usuario de AWS CloudFormation .

Ejemplos

WebSocket API simple

El siguiente ejemplo crea una WebSocket API con tres rutas.

```
Resources:
  MyWebSocketApi:
    Type: AWS::Serverless::WebSocketApi
    Properties:
      RouteSelectionExpression: $request.body.action
      Routes:
        $connect:
          FunctionArn: !GetAtt ConnectFunction.Arn
        $disconnect:
          FunctionArn: !GetAtt DisconnectFunction.Arn
        sendMessage:
          FunctionArn: !GetAtt SendMessageFunction.Arn

  ConnectFunction:
    Type: AWS::Serverless::Function
    Properties:
      Handler: index.connect
      Runtime: nodejs20.x
      CodeUri: ./src

  DisconnectFunction:
    Type: AWS::Serverless::Function
    Properties:
      Handler: index.disconnect
      Runtime: nodejs20.x
      CodeUri: ./src

  SendMessageFunction:
    Type: AWS::Serverless::Function
    Properties:
      Handler: index.sendMessage
      Runtime: nodejs20.x
      CodeUri: ./src
```

WebSocket API con Lambda Authorizer

En el siguiente ejemplo, se crea una WebSocket API con un autorizador Lambda.

```
Resources:
  MyWebSocketApi:
    Type: AWS::Serverless::WebSocketApi
    Properties:
      RouteSelectionExpression: $request.body.action
```

```
Auth:
  AuthType: CUSTOM
  AuthArn: !GetAtt AuthorizerFunction.Arn
  IdentitySource:
    - route.request.header.Authorization
Routes:
  $connect:
    FunctionArn: !GetAtt ConnectFunction.Arn
  sendMessage:
    FunctionArn: !GetAtt SendMessageFunction.Arn
```

```
AuthorizerFunction:
  Type: AWS::Serverless::Function
  Properties:
    Handler: index.authorize
    Runtime: nodejs20.x
    CodeUri: ./src
```

```
ConnectFunction:
  Type: AWS::Serverless::Function
  Properties:
    Handler: index.connect
    Runtime: nodejs20.x
    CodeUri: ./src
```

```
SendMessageFunction:
  Type: AWS::Serverless::Function
  Properties:
    Handler: index.sendMessage
    Runtime: nodejs20.x
    CodeUri: ./src
```

WebSocketApiAuth

Configure la autorización para controlar el acceso a su API de Amazon WebSocket API Gateway.

Para obtener más información sobre la configuración del acceso a WebSocket las API, consulte [Control del acceso a WebSocket las API](#) en la Guía para desarrolladores de API Gateway.

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
AuthArn: String
AuthType: String
IdentitySource: List
InvokeRole: String
Name: String
```

Propiedades

AuthArn

El ARN de la función de eventos de Lambda que se utilizará para la autorización. Se requiere cuando AuthType es. CUSTOM

Tipo: cadena

Obligatorio: condicional

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

AuthType

El tipo de autorización. Los valores válidos son NONE, AWS_IAM o CUSTOM.

- NONE- Sin autorización
- AWS_IAM- Autorización de IAM
- CUSTOM- Autorizador Lambda

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

IdentitySource

Es la fuente de identidad para la que se solicita autorización. Por ejemplo, `route.request.header.Authorization`.

Tipo: lista

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [IdentitySource](#) propiedad de un `AWS::ApiGatewayV2::Authorizer` recurso.

InvokeRole

El ARN de la función de IAM que asume API Gateway al invocar la función de autorización. Si no se especifica, crea AWS SAM automáticamente un permiso basado en recursos que permite a API Gateway invocar la función de autorización.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [AuthorizerCredentialsArn](#) propiedad de un recurso. `AWS::ApiGatewayV2::Authorizer`

Name

El nombre del autorizador.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Name](#) propiedad de un `AWS::ApiGatewayV2::Authorizer` recurso.

Ejemplos

Autorizador Lambda

El siguiente ejemplo configura un autorizador Lambda para una API. WebSocket

```
Auth:
  AuthType: CUSTOM
  AuthArn: !GetAtt AuthorizerFunction.Arn
  IdentitySource:
    - route.request.header.Authorization
```

Autorización de IAM

El siguiente ejemplo configura la autorización de IAM para una API. WebSocket

```
Auth:  
  AuthType: AWS_IAM
```

WebSocketApiDomainConfiguration

Configura un dominio personalizado para una WebSocket API.

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
BasePath: List  
CertificateArn: String  
DomainName: String  
EndpointConfiguration: String  
Route53: Route53Configuration  
SecurityPolicy: String
```

Propiedades

BasePath

Una lista de las rutas base que se deben configurar con el nombre de dominio de Amazon API Gateway.

Tipo: lista

Obligatorio: no

Valor predeterminado: /

CloudFormation compatibilidad: esta propiedad es similar a la [ApiMappingKey](#) propiedad de un `AWS::ApiGatewayV2::ApiMapping` recurso. AWS SAM crea varios `AWS::ApiGatewayV2::ApiMapping` recursos, uno por cada valor especificado en esta propiedad.

CertificateArn

El nombre de recurso de Amazon (ARN) de un certificado AWS administrado para el punto final de este nombre de dominio. AWS Gestor de certificados es la única fuente compatible.

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [CertificateArn](#) propiedad de un `AWS::ApiGatewayV2::DomainName` `DomainNameConfiguration` recurso.

DomainName

El nombre de dominio personalizado de tu API WebSocket API Gateway. No se admiten letras mayúsculas.

AWS SAM genera un `AWS::ApiGatewayV2::DomainName` recurso cuando se establece esta propiedad. Para obtener información sobre CloudFormation los recursos generados, consulte [Generado CloudFormation recursos para AWS SAM](#).

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [DomainName](#) propiedad de un `AWS::ApiGatewayV2::DomainName` recurso.

EndpointConfiguration

Define el tipo de punto de conexión de API Gateway que se va a asignar al dominio personalizado. El valor de esta propiedad determina cómo se mapea la `CertificateArn` propiedad. CloudFormation

El único valor válido para WebSocket APIs es `REGIONAL`.

Tipo: cadena

Obligatorio: no

Valor predeterminado: `REGIONAL`

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

Route53

Define una configuración de Route 53.

Tipo: [Route53Configuration](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

SecurityPolicy

La versión TLS de la política de seguridad para este nombre de dominio.

El único valor válido para WebSocket APIs es TLS_1_2.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [SecurityPolicy](#) propiedad del tipo de AWS::ApiGatewayV2::DomainName DomainNameConfiguration datos.

Ejemplos

DomainName

DomainName ejemplo

YAML

```
Domain:
  DomainName: ws.example.com
  CertificateArn: arn:aws:acm:us-east-1:123456789012:certificate/example
  EndpointConfiguration: REGIONAL
  Route53:
    HostedZoneId: Z1PA6795UKMFR9
  BasePath:
    - v1
    - v2
```

Route53Configuration

Configura los conjuntos de registros de Route53 para una API.

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
EvaluateTargetHealth: Boolean  
HostedZoneId: String  
HostedZoneName: String  
IPv6: Boolean  
Region: String  
SetIdentifier: String
```

Propiedades

EvaluateTargetHealth

Si EvaluateTargetHealth es verdadero, un registro de alias hereda el estado del AWS recurso al que se hace referencia, como un balanceador de cargas de Elastic Load Balancing u otro registro de la zona alojada.

Tipo: Booleano

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [EvaluateTargetHealth](#) propiedad de un `AWS::Route53::RecordSetGroup` AliasTarget recurso.

Notas adicionales: No se puede EvaluateTargetHealth establecer en true cuando el alias objetivo es una CloudFront distribución.

HostedZoneId

ID de la zona alojada en la que desea crear registros.

Especifica HostedZoneName o HostedZoneId, pero no ambos. Si tiene múltiples zonas hospedadas con el mismo nombre de dominio, debe especificar la zona hospedada mediante HostedZoneId.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [HostedZoneId](#) propiedad de un `AWS::Route53::RecordSetGroup` `RecordSet` recurso.

HostedZoneName

El nombre de la zona alojada en la que desea crear registros. Debe incluir un punto final (por ejemplo, `www.example.com.`) como parte del `HostedZoneName`.

Especifica `HostedZoneName` o `HostedZoneId`, pero no ambos. Si tiene múltiples zonas hospedadas con el mismo nombre de dominio, debe especificar la zona hospedada mediante `HostedZoneId`.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [HostedZoneName](#) propiedad de un `AWS::Route53::RecordSetGroup` `RecordSet` recurso.

IPv6

Cuando se establece esta propiedad, AWS SAM crea un `AWS::Route53::RecordSet` recurso y establece [Type](#) en AAAA el valor proporcionado `HostedZone`.

Tipo: Booleano

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

Region

Solo conjuntos de registros de recursos de latencia: La región de Amazon EC2 donde haya creado el recurso al que se refiere este conjunto de registros de recursos. El recurso suele ser un AWS recurso, como una instancia EC2 o un balanceador de cargas ELB, y se hace referencia a él mediante una dirección IP o un nombre de dominio DNS, según el tipo de registro.

Cuando Amazon Route 53 recibe una consulta de DNS para un nombre de dominio y tipo para los que ha creado conjuntos de registros de recursos de latencia, Route 53 selecciona el conjunto

de registros de recursos de latencia con la menor latencia entre el usuario final y la región de Amazon EC2 asociada. Route 53 devuelve a continuación el valor asociado con el conjunto de registros de recursos seleccionados.

Tenga en cuenta lo siguiente:

- Solo puede especificar un `ResourceRecord` por conjunto de registros de recursos de latencia.
- Solo puede crear un conjunto de registros de recursos de latencia para cada región de Amazon EC2.
- No es necesario que cree conjuntos de registros de recursos de latencia para todas las regiones de Amazon EC2. Route 53 elige la región con la mejor latencia entre las regiones para las que crea conjuntos de registros de recursos de latencia.
- No puede crear conjuntos de registros de recursos sin latencia que tengan los mismos valores `Name` y `Type` que los conjuntos de registros de recursos de latencia.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [Region](#) propiedad de un `AWS::Route53::RecordSetGroup RecordSet` tipo de datos.

SetIdentifier

Conjuntos de registros de recursos que tienen una política de direccionamiento que no es simple: Un identificador que diferencia entre varios conjuntos de registros de recursos con la misma combinación de nombre y tipo, como varios conjuntos de registros de recursos ponderados denominados `acme.example.com` de tipo A. En un grupo de conjuntos de registros de recursos que tengan el mismo nombre y tipo, el valor de `SetIdentifier` debe ser exclusivo para cada conjunto de registros de recursos.

Para obtener información sobre cómo dirigir las políticas, consulta [Elección de una política de direccionamiento](#) en la Guía para desarrolladores de Amazon Route 53.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [SetIdentifier](#) propiedad de un tipo de `AWS::Route53::RecordSetGroup RecordSet` datos.

Ejemplos

Ejemplo de configuración de Route 53

En este ejemplo se muestra cómo configurar Route 53.

YAML

```
Domain:
  DomainName: www.example.com
  CertificateArn: arn-example
  EndpointConfiguration: REGIONAL
  Route53:
    HostedZoneId: Z1PA6795UKMFR9
    EvaluateTargetHealth: true
```

RouteConfiguration

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
ApiKeyRequired: Boolean
FunctionArn: String
IntegrationTimeout: Integer
ModelSelectionExpression: String
OperationName: String
RequestModels: Json
RequestParameters: Json
RouteResponseSelectionExpression: String
```

Propiedades

ApiKeyRequired

Especifica si se requiere una clave de API para esta ruta.

Tipo: Booleano

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [ApiKeyRequired](#) propiedad de un `AWS::ApiGatewayV2::Route` recurso.

FunctionArn

El ARN de la función de eventos de Lambda que gestiona esta ruta. AWS SAM crea una integración y los permisos necesarios para que API Gateway invoque la función.

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

IntegrationTimeout

El tiempo de espera de la integración, en milisegundos. El valor máximo es de 29 000 milisegundos (29 segundos).

Tipo: entero

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [TimeoutInMillis](#) propiedad de un `AWS::ApiGatewayV2::Integration` recurso.

ModelSelectionExpression

Es la expresión de selección del modelo para la ruta.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [ModelSelectionExpression](#) propiedad de un `AWS::ApiGatewayV2::Route` recurso.

OperationName

Es el nombre de la operación para la ruta.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [OperationName](#) propiedad de un `AWS::ApiGatewayV2::Route` recurso.

RequestModels

Son los modelos de solicitud para la ruta.

Tipo: Json

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [RequestModels](#) propiedad de un `AWS::ApiGatewayV2::Route` recurso.

RequestParameters

Son los parámetros de solicitud para la ruta.

Tipo: Json

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [RequestParameters](#) propiedad de un `AWS::ApiGatewayV2::Route` recurso.

RouteResponseSelectionExpression

Es la expresión de selección de respuesta de la ruta para la ruta.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [RouteResponseSelectionExpression](#) propiedad de un `AWS::ApiGatewayV2::Route` recurso.

Ejemplos

Ruta sencilla

El siguiente ejemplo configura una ruta sencilla.

Routes :

```
$connect:
  FunctionArn: !GetAtt ConnectFunction.Arn
sendMessage:
  FunctionArn: !GetAtt SendMessageFunction.Arn
  OperationName: SendMessage
```

AWS::Serverless::LayerVersion

Crea una Lambda LayerVersion que contiene el código de biblioteca o de tiempo de ejecución que necesita una función de eventos de Lambda.

El [AWS::Serverless::LayerVersion](#) recurso también admite el atributo `Metadata resource`, por lo que puede indicarle que cree AWS SAM las capas incluidas en su aplicación. Para obtener más información sobre la creación de capas, consulta [Creación de capas Lambda en AWS SAM](#).

Nota importante: Desde la publicación del atributo de [UpdateReplacePolicy](#) recurso en CloudFormation, [AWS::Lambda::LayerVersion](#) (recomendado) ofrece las mismas ventajas que [AWS::Serverless::LayerVersion](#).

Cuando se transforma un servidor sin servidor LayerVersion, SAM también transforma el identificador lógico del recurso para que los antiguos no LayerVersions se eliminen automáticamente CloudFormation cuando se actualice el recurso.

Note

Cuando lo implementa AWS CloudFormation, AWS SAM transforma sus AWS SAM recursos en CloudFormation recursos. Para obtener más información, consulte [Generado CloudFormation recursos para AWS SAM](#).

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
Type: AWS::Serverless::LayerVersion
Properties:
```

```
CompatibleArchitectures: List  
CompatibleRuntimes: List  
ContentUri: String / LayerContent  
Description: String  
LayerName: String  
LicenseInfo: String  
PublishLambdaVersion: Boolean  
RetentionPolicy: String
```

Propiedades

CompatibleArchitectures

Especifica las arquitecturas del conjunto de instrucciones compatibles para la versión de capa.

Para obtener más información sobre esta propiedad, consulta [Arquitecturas del conjunto de instrucciones Lambda](#) en la Guía para desarrolladores de AWS Lambda .

Valores válidos: x86_64, arm64

Tipo: lista

Obligatorio: no

Valor predeterminado: x86_64

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [CompatibleArchitectures](#) propiedad de un `AWS::Lambda::LayerVersion` recurso.

CompatibleRuntimes

Lista de tiempos de ejecución compatibles con esto LayerVersion.

Tipo: lista

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [CompatibleRuntimes](#) propiedad de un `AWS::Lambda::LayerVersion` recurso.

ContentUri

Uri de Amazon S3, ruta a la carpeta local u LayerContent objeto del código de capa.

Si se proporciona un URI o un `LayerContent` objeto de Amazon S3, el objeto de Amazon S3 al que se hace referencia debe ser un archivo ZIP válido que contenga el contenido de una [capa Lambda](#).

Si se proporciona una ruta a una carpeta local, para que el contenido se transforme correctamente, la plantilla debe seguir el flujo de trabajo que incluye [compilación de sam](#) seguido de [sam deploy](#) o [sam package](#). De forma predeterminada, las rutas relativas se resuelven con respecto a la ubicación de la AWS SAM plantilla.

Tipo: Cadena | [LayerContent](#)

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad es similar a la [Content](#) propiedad de un `AWS::Lambda::LayerVersion` recurso. Las propiedades anidadas de Amazon S3 tienen un nombre diferente.

Description

Descripción de esta capa.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Description](#) propiedad de un `AWS::Lambda::LayerVersion` recurso.

LayerName

El nombre o nombre de recurso de Amazon (ARN) de la capa.

Tipo: cadena

Obligatorio: no

Predeterminado: identificador lógico del recurso

CloudFormation compatibilidad: esta propiedad es similar a la [LayerName](#) propiedad de un `AWS::Lambda::LayerVersion` recurso. Si no especificas un nombre, se utilizará como nombre el ID lógico del recurso.

LicenseInfo

Información sobre la licencia correspondiente `LayerVersion`.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [LicenseInfo](#) propiedad de un `AWS::Lambda::LayerVersion` recurso.

`PublishLambdaVersion`

Propiedad opcional que crea una nueva versión de Lambda cada vez que se produce un cambio en el recurso `LayerVersion` al que se hace referencia. Al habilitar la función de Lambda conectada con `AutoPublishAlias` y `AutoPublishAliasAllProperties`, se creará una nueva versión de Lambda para cada cambio realizado en el recurso `LayerVersion`.

Tipo: Booleano

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

`RetentionPolicy`

Esta propiedad especifica si las versiones antiguas de su `LayerVersion` se retienen o se eliminan al eliminar un recurso. Si tiene que retener sus versiones antiguas de `LayerVersion` al actualizar o reemplazar un recurso, debe tener el atributo `UpdateReplacePolicy` activado. Para obtener información sobre cómo hacerlo, consulte el [Atributo UpdateReplacePolicy](#) en la Guía del usuario de AWS CloudFormation .

Valores válidos: `Retain` o `Delete`

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

Notas adicionales: Al especificar `Retain`, AWS SAM agrega un [Atributos de recursos compatibles con AWS SAM](#) valor de `DeletionPolicy`: `Retain` al `AWS::Lambda::LayerVersion` recurso transformado.

valores devueltos

Ref.

Cuando se proporciona el identificador lógico de este recurso a la función Ref intrínseca, devuelve el ARN del recurso de la Lambda subyacente. LayerVersion

Para obtener más información sobre el uso de la función Ref, consulta [Ref](#) en la Guía del usuario de AWS CloudFormation .

Ejemplos

LayerVersionExample

Ejemplo de un LayerVersion

YAML

```
Properties:
  LayerName: MyLayer
  Description: Layer description
  ContentUri: 's3://sam-s3-demo-bucket/my-layer.zip'
  CompatibleRuntimes:
    - nodejs10.x
    - nodejs12.x
  LicenseInfo: 'Available under the MIT-0 license.'
  RetentionPolicy: Retain
```

LayerContent

Archivo ZIP que incluye el contenido de una [capa de Lambda](#).

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
Bucket: String
Key: String
StorageMode: String
```

Version: *String*

Propiedades

Bucket

El bucket de Amazon S3 del archivo de la capa.

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [S3Bucket](#) propiedad del tipo de `AWS::Lambda::LayerVersion` Content datos.

Key

La clave de Amazon S3 del archivo de la capa.

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [S3Key](#) propiedad del tipo de `AWS::Lambda::LayerVersion` Content datos.

StorageMode

Controla la forma en que Lambda almacena el archivo de capas.

- **COPY**— Lambda carga una copia de su archivo de capas al almacenamiento. Lambda-managed Puede eliminar el objeto de Amazon S3 después de que Lambda cree la versión de capa.
- **REFERENCE**— Lambda hace referencia al archivo de capas de su bucket de Amazon S3 y no almacena ninguna copia. El objeto debe permanecer en su lugar y Lambda debe conservar el acceso a él durante toda la vigencia de la versión de la capa.

Para usarloREFERENCE, habilite el control de versiones en su bucket de Amazon S3 y conceda al servicio de Lambda acceso principal al objeto. Para obtener más información, consulte el almacenamiento de código de [Self-managed Amazon S3](#) en la Guía para AWS Lambda desarrolladores.

Tipo: cadena

Valores válidos: COPY | REFERENCE

Obligatorio: no

Valor predeterminado: COPY

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [S3ObjectStorageMode](#) propiedad del tipo de AWS::Lambda::LayerVersion Content datos.

Version

Para objetos con control de versiones, el objeto de archivo de la capa que se va a utilizar.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [S3ObjectVersion](#) propiedad del tipo de AWS::Lambda::LayerVersion Content datos.

Ejemplos

LayerContent

Ejemplo de contenido de capa

YAML

```
LayerContent:
  Bucket: amzn-s3-demo-bucket-name
  Key: mykey-name
  Version: 121212
```

Self-managed Almacenamiento de código en Amazon S3

A continuación se LayerContent hace referencia al archivo de capas de su propio bucket de Amazon S3 en lugar de copiarlo para Lambda-managed almacenarlo.

YAML

```
LayerContent:
```

```
Bucket: amzn-s3-demo-bucket-name
Key: mykey-name
Version: 121212
StorageMode: REFERENCE
```

AWS::Serverless::MicrovmImage

Crea una imagen de AWS Lambda microVM a partir de un artefacto de código almacenado en Amazon S3. Una imagen de microVM es una Firecracker instantánea creada a partir del código de una aplicación Dockerfile y que permite el inicio en menos de un segundo de las instancias de microVM.

Note

Al realizar la implementación en AWS CloudFormation, AWS SAM transforma AWS SAM los recursos en recursos. CloudFormation Para obtener más información, consulte [Generado CloudFormation recursos para AWS SAM](#).

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
Type: AWS::Serverless::MicrovmImage
Properties:
  AdditionalOsCapabilities: List
  BaseImageArn: String
  BaseImageVersion: String
  BuildRoleArn: String
  CodeUri: String
  CpuConfigurations: List
  Description: String
  EgressNetworkConnectors: List
  EnvironmentVariables: Map
  Hooks: Hooks
  Logging: Logging
  Name: String
```

[PropagateTags](#): *Boolean*
[Resources](#): *List*
[Tags](#): *Map*

Propiedades

`AdditionalOsCapabilities`

Capacidades de sistema operativo adicionales que se pueden conceder al entorno de ejecución de MicroVM. Si no especificas un valor, el servicio no otorga capacidades de sistema operativo adicionales.

Valores válidos: ALL

Tipo: lista

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [AdditionalOsCapabilities](#) propiedad de un `AWS::Lambda::MicrovmImage` recurso.

`BaseImageArn`

El ARN de la imagen microVM base a partir de la cual se va a construir.

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [BaseImageArn](#) propiedad de un `AWS::Lambda::MicrovmImage` recurso.

`BaseImageVersion`

La versión de la imagen microVM base que se va a utilizar.

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [BaseImageVersion](#) propiedad de un `AWS::Lambda::MicrovmImage` recurso.

BuildRoleArn

El ARN de la función de IAM que asume el servicio de compilación de microVM para descargar el artefacto de código de Amazon S3 y escribir los registros de compilación.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es similar a la [BuildRoleArn](#) propiedad de un recurso. `AWS::Lambda::MicrovmImage` Esto es obligatorio en CloudFormation , pero no en AWS SAM. Si no especifica un rol, AWS SAM crea uno con un identificador lógico de `<resource-logical-id>BuildRole`. Si tu artefacto de código está cifrado con una clave gestionada por el cliente AWS KMS, debes proporcionar tu propia función con el `kms:Decrypt` permiso correspondiente.

CodeUri

El URI de Amazon S3 del artefacto zip que contiene el código de la aplicación Dockerfile y.

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se transforma en la [CodeArtifact](#) propiedad de un `AWS::Lambda::MicrovmImage` recurso. AWS SAM envuelve el URI en la `CodeArtifact.Uri` estructura.

CpuConfigurations

Una lista de [CpuConfiguration](#) objetos que especifican las arquitecturas de CPU compatibles con la microVM. Si no especificas un valor, el servicio utiliza la arquitectura de CPU predeterminada.

Tipo: lista

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [CpuConfigurations](#) propiedad de un `AWS::Lambda::MicrovmImage` recurso.

Description

Descripción de la imagen de la microVM.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Description](#) propiedad de un `AWS::Lambda::MicrovmImage` recurso.

EgressNetworkConnectors

La lista de ARN de conectores de red de salida disponibles para la microVM en tiempo de ejecución.

Tipo: lista

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [EgressNetworkConnectors](#) propiedad de un recurso. `AWS::Lambda::MicrovmImage`

EnvironmentVariables

Variables de entorno configuradas en el entorno de ejecución de MicroVM.

Tipo: mapa

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es similar a la [EnvironmentVariables](#) propiedad de un `AWS::Lambda::MicrovmImage` recurso. En AWS SAM, especifique como un mapa (por ejemplo, `KEY: value`). AWS SAM convierte al formato CloudFormation de matriz `[{Key: KEY, Value: value}]`.

Hooks

Configuración de enlaces durante el ciclo de vida para imágenes de micromáquinas virtuales y micromáquinas virtuales.

[Tipo: enlaces](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [Hooks](#) propiedad de un recurso. `AWS::Lambda::MicrovmImage`

Logging

Configuración para la salida de registro de microVM. Especifique exactamente una: CloudWatch habilitar el CloudWatch registro o Disabled desactivarlo.

Type: [Logging](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [Logging](#) propiedad de un `AWS::Lambda::MicrovmImage` recurso.

Name

El nombre de la imagen de la microVM.

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Name](#) propiedad de un `AWS::Lambda::MicrovmImage` recurso.

PropagateTags

Especifica si se deben pasar etiquetas de la Tags propiedad a los recursos generados, como el rol de compilación generado automáticamente.

Tipo: Booleano

Obligatorio: no

Valor predeterminado: False

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

Resources

Una lista de objetos de [Resources](#) que especifican los requisitos de recursos de la microVM.

Tipo: lista

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [Resources](#) propiedad de un `AWS::Lambda::MicrovmImage` recurso.

Tags

Un mapa de pares clave-valor que especifica las etiquetas agregadas a esta imagen de microVM.

Tipo: mapa

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es similar a la [Tags](#) propiedad de un recurso. `AWS::Lambda::MicrovmImage` La `Tags` propiedad en se AWS SAM compone de pares clave-valor (mientras que en CloudFormation esta propiedad se compone de una lista de Tag objetos). Además, agrega AWS SAM automáticamente una `lambda:createdBy:SAM` etiqueta a esta imagen de microVM y a las funciones predeterminadas que se generan para ella.

valores devueltos

Ref.

Cuando se proporciona el ID lógico de este recurso a la función `Ref` intrínseca, devuelve el ARN de la imagen de la microVM.

Para obtener más información sobre el uso de la función `Ref`, consulta [Ref](#) en la Guía del usuario de AWS CloudFormation .

Ventilador:: `GetAtt`

`Fn::GetAtt` devuelve un valor para un atributo especificado de este tipo. A continuación se indican los atributos disponibles y los valores devueltos de muestra.

Para obtener más información acerca de cómo usar `Fn::GetAtt`, consulta [Fn::GetAtt](#) en la Guía del usuario de AWS CloudFormation .

`CreatedAt`

La marca de tiempo en que se creó la versión de imagen de microVM.

`ImageArn`

El ARN de la imagen de la microVM.

LatestActiveImageVersion

La última versión activa de la imagen de la microVM.

LatestFailedImageVersion

La última versión fallida de la imagen de la microVM, si la hay.

State

El estado actual de la imagen de la microVM.

UpdatedAt

La marca de tiempo en la que se actualizó por última vez la versión de la imagen de la microVM.

Ejemplos

Imagen microVM mínima

El siguiente ejemplo crea una imagen de microVM con solo las propiedades necesarias. AWS SAM genera automáticamente un rol de compilación.

```
AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31

Resources:
  MyMicroVM:
    Type: AWS::Serverless::MicrovmImage
    Properties:
      Name: my-app
      CodeUri: s3://my-bucket/app.zip
      BaseImageArn: arn:aws:lambda:us-east-2:aws:microvm-image:a12023-1
      BaseImageVersion: "0"
```

Configuración completa de imágenes de microVM

El siguiente ejemplo crea una imagen de microVM con todas las propiedades disponibles configuradas.

```
AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31
```

```

Resources:
  MyMicroVM:
    Type: AWS::Serverless::MicrovmImage
    Properties:
      Name: my-app
      CodeUri: s3://my-bucket/app.zip
      BaseImageArn: arn:aws:lambda:us-east-2:aws:microvm-image:al2023-1
      BaseImageVersion: "0"
      Description: "Production MicroVM application"
      CpuConfigurations:
        - Architecture: ARM_64
      Resources:
        - MinimumMemoryInMiB: 1024
      AdditionalOsCapabilities:
        - ALL
      EgressNetworkConnectors:
        - arn:aws:lambda:us-east-2:aws:network-connector:aws-network-
connector:INTERNET_EGRESS
      Hooks:
        Port: 9000
        MicrovmHooks:
          Run: ENABLED
          RunTimeoutInSeconds: 30
          Suspend: ENABLED
          Resume: ENABLED
        MicrovmImageHooks:
          Ready: ENABLED
          ReadyTimeoutInSeconds: 60
      EnvironmentVariables:
        LOG_LEVEL: info
        APP_ENV: production
      Logging:
        CloudWatch:
          LogGroup: /aws/lambda-microvms/my-app
      Tags:
        Environment: Production
        Team: Platform

```

AWS::Serverless::NetworkConnector

Crea un conector AWS Lambda de red que define la conectividad de red para los recursos informáticos de Lambda. Un conector de red especifica cómo se conectan los recursos informáticos a las redes, por ejemplo, mediante el enrutamiento del tráfico de salida a través de una VPC del

cliente mediante interfaces de red elásticas (ENI) que el servicio aprovisiona en subredes y grupos de seguridad específicos. Cada conector sirve para un tipo de procesamiento, ya sea microVMS o Web Functions. Para usar un conector con un [AWS::Serverless::WebFunction](#), `ComputeTypes` defina `WebFunction` y pase el conector `LatestVersionArn` a la propiedad de la función.

`EgressNetworkConnectorArn`

Note

Cuando despliegas en AWS CloudFormation, AWS SAM transforma tus AWS SAM recursos en CloudFormation recursos. Para obtener más información, consulte [Generado CloudFormation recursos para AWS SAM](#).

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
Type: AWS::Serverless::NetworkConnector
Properties:
  ComputeTypes: List
  Name: String
  OperatorRole: String
  PropagateTags: Boolean
  Tags: Map
  VpcConfig: VpcConfig
```

Propiedades

ComputeTypes

El tipo de recurso informático Lambda que puede utilizar este conector de red. Especifique exactamente uno de los siguientes valores:

- `MicroVm`: Los recursos informáticos de MicroVM utilizan el conector.
- `WebFunction`: [Los WebFunction recursos utilizan el AWS::Serverless conector a través de su `EgressNetworkConnectorArn` propiedad.](#)

Tipo: lista

Obligatorio: no

Valor predeterminado: [MicroVm]

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [Configuration.VpcEgressConfiguration.AssociatedComputeResourceTypes](#) propiedad de un `AWS::Lambda::NetworkConnector` recurso.

Name

El nombre del conector de red. El nombre debe ser único en su cuenta y región.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [Name](#) propiedad de un `AWS::Lambda::NetworkConnector` recurso.

OperatorRole

El ARN de la función de IAM que asume el servicio de conector de red para crear y administrar interfaces de red elásticas (ENI).

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es similar a la [OperatorRole](#) propiedad de un recurso `AWS::Lambda::NetworkConnector`. Esto es obligatorio en CloudFormation, pero no en AWS SAM. Si no especifica un rol, AWS SAM crea uno con un identificador lógico de `<resource-logical-id>OperatorRole`.

PropagateTags

Especifica si se deben pasar etiquetas de la `Tags` propiedad a los recursos generados, como el rol de operador generado automáticamente.

Tipo: Booleano

Obligatorio: no

Valor predeterminado: False

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

Tags

Un mapa de pares clave-valor que especifica las etiquetas agregadas a este conector de red.

Tipo: mapa

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es similar a la [Tags](#) propiedad de un `AWS::Lambda::NetworkConnector` recurso. La `Tags` propiedad en se AWS SAM compone de pares clave-valor (mientras que en CloudFormation esta propiedad se compone de una lista de Tag objetos). Además, agrega AWS SAM automáticamente una `lambda:createdBy:SAM` etiqueta a este conector de red y a las funciones predeterminadas que se generan para él.

VpcConfig

La configuración de salida de VPC para el conector de red. Especifica las subredes, los grupos de seguridad y el protocolo de red para enrutar el tráfico saliente a través de la VPC.

Tipo: [VpcConfig](#)

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se transforma en la [Configuration.VpcEgressConfiguration](#) propiedad de un recurso. `AWS::Lambda::NetworkConnector` AWS SAM conjuntos `AssociatedComputeResourceTypes` de la [ComputeTypes](#) propiedad.

valores devueltos

Ref.

Cuando se proporciona el identificador lógico de este recurso a la función `Ref` intrínseca, devuelve el ARN del conector de red.

Para obtener más información sobre el uso de la función `Ref`, consulta [Ref](#) en la Guía del usuario de AWS CloudFormation .

Ventilador:: `GetAtt`

`Fn::GetAtt` devuelve un valor para un atributo especificado de este tipo. A continuación se indican los atributos disponibles y los valores devueltos de muestra.

Para obtener más información acerca de cómo usar `Fn::GetAtt`, consulta [Fn::GetAtt](#) en la Guía del usuario de AWS CloudFormation .

Arn

El nombre de recurso de Amazon (ARN) del conector de red.

LatestVersionArn

El ARN compatible con la versión de la última versión del conector de red, por ejemplo.

`arn:aws:lambda:us-east-1:123456789012:network-connector:my-connector:1`

[Utilice este valor para la `EgressNetworkConnectorArn` propiedad de un `AWS::Serverless::WebFunction`](#)

State

El estado actual del conector de red. Los valores válidos son PENDING, ACTIVE, INACTIVE, FAILED, DELETING y DELETE_FAILED.

Ejemplos

Conector de red básico con función generada automáticamente

En el siguiente ejemplo, se crea un conector de red de salida de VPC. AWS SAM genera automáticamente el rol de operador.

```
AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31

Resources:
  MyVpcConnector:
    Type: AWS::Serverless::NetworkConnector
    Properties:
      Name: my-vpc-connector
      VpcConfig:
        SubnetIds:
          - subnet-0abc123def4567890
          - subnet-0def456abc7890123
        SecurityGroupIds:
          - sg-0abc1234abcde0001
        NetworkProtocol: IPv4
      Tags:
        Environment: Production
```

Conector de red para una función web

El siguiente ejemplo crea un conector de red para Web Functions y lo utiliza para el tráfico saliente de una función web.

```
AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31

Resources:
  MyWebFunctionConnector:
    Type: AWS::Serverless::NetworkConnector
    Properties:
      ComputeTypes:
        - WebFunction
      VpcConfig:
        SubnetIds:
          - subnet-0abc123def4567890
        SecurityGroupIds:
          - sg-0abc1234abcde0001
        NetworkProtocol: IPv4

  MyWebFunction:
    Type: AWS::Serverless::WebFunction
    Properties:
      Runtime: nodejs24.x
      AuthType: IamAuth
      CodeUri: s3://amzn-s3-demo-bucket/webfn.zip
      EgressNetworkConnectorArn: !GetAtt MyWebFunctionConnector.LatestVersionArn
```

VpcConfig

Configura los ajustes de salida de la VPC para un conector de red, incluidas las subredes, los grupos de seguridad y el protocolo de red para el aprovisionamiento de la interface elastic network (ENI).

Sintaxis

Para declarar esta entidad en su plantilla AWS Serverless Application Model (AWS SAM), utilice la siguiente sintaxis.

YAML

```
NetworkProtocol: String
```

[SecurityGroupIds](#): *List*

[SubnetIds](#): *List*

Propiedades

NetworkProtocol

El protocolo de red para la conexión de salida de la VPC.

Valores válidos: IPv4, DualStack

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [NetworkProtocol](#) propiedad Configuration.VpcEgressConfiguration de un AWS::Lambda::NetworkConnector recurso.

SecurityGroupIds

Una lista de identificadores de grupos de seguridad para asociarlos a las interfaces de red elásticas (ENI).

Tipo: lista

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [SecurityGroupIds](#) propiedad Configuration.VpcEgressConfiguration de un AWS::Lambda::NetworkConnector recurso.

SubnetIds

Una lista de identificadores de subred en los que el servicio aprovisiona ENI para la conectividad de salida de la VPC.

Tipo: lista

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [SubnetIds](#) propiedad de Configuration.VpcEgressConfiguration un recurso. AWS::Lambda::NetworkConnector

Ejemplos

VpcConfig ejemplo

```
VpcConfig:
  SubnetIds:
    - subnet-0abc123def4567890
    - subnet-0def456abc7890123
  SecurityGroupIds:
    - sg-0abc1234def567890
  NetworkProtocol: IPv4
```

AWS::Serverless::SimpleTable

Crea una tabla de DynamoDB con una clave principal de atributo único. Resulta útil cuando solo es necesario acceder a los datos mediante una clave principal.

Para obtener características más avanzadas, utilice un recurso [AWS::DynamoDB::Table](#) en CloudFormation. Estos recursos se pueden utilizar en AWS SAM. Son completos y proporcionan una mayor personalización, como [key schema](#) y [resource policy](#).

Note

Cuando despliega en AWS CloudFormation, AWS SAM transforma sus AWS SAM recursos en CloudFormation recursos. Para obtener más información, consulte [Generado CloudFormation recursos para AWS SAM](#).

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
Type: AWS::Serverless::SimpleTable
Properties:
  PointInTimeRecoverySpecification: PointInTimeRecoverySpecification
  PrimaryKey: PrimaryKeyObject
```

ProvisionedThroughput: [*ProvisionedThroughputObject*](#)
SSESpecification: [*SSESpecification*](#)
TableName: *String*
Tags: *Map*

Propiedades

PointInTimeRecoverySpecification

La configuración utilizada para habilitar la recuperación a un momento dado.

Tipo: [PointInTimeRecoverySpecification](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [PointInTimeRecoverySpecification](#) propiedad de un `AWS::DynamoDB::Table` recurso.

PrimaryKey

Nombre y tipo de atributo que se utilizará como clave principal de la tabla. Si no se proporciona, la clave principal será `String` con un valor de `id`.

Note

El valor de esta propiedad no se puede modificar después de crear este recurso.

Tipo: [PrimaryKeyObject](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

ProvisionedThroughput

Lea y escriba la información de aprovisionamiento de rendimiento.

Si `ProvisionedThroughput` no se especifica, `BillingMode` se especificará como `PAY_PER_REQUEST`.

Tipo: [ProvisionedThroughputObject](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [ProvisionedThroughput](#) propiedad de un `AWS::DynamoDB::Table` recurso.

SSESpecification

Especifica la configuración para habilitar el cifrado del lado del servidor.

Tipo: [SSESpecification](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [SSESpecification](#) propiedad de un `AWS::DynamoDB::Table` recurso.

TableName

Nombre de la tabla de DynamoDB.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [TableName](#) propiedad de un `AWS::DynamoDB::Table` recurso.

Tags

Un mapa (cadena a cadena) que especifica las etiquetas que se van a añadir a este SimpleTable. Para obtener más información sobre las claves y los valores válidos de las etiquetas, consulta la [etiqueta de recursos](#) en la Guía del usuario de AWS CloudFormation .

Tipo: mapa

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es similar a la [Tags](#) propiedad de un `AWS::DynamoDB::Table` recurso. La propiedad Tags de SAM consta de pares clave-valor; en CloudFormation ella, consiste en una lista de objetos Tag.

valores devueltos

Ref.

Cuando se proporciona el ID lógico de este recurso a la función intrínseca Ref, esta devuelve el nombre del recurso de la función de Lambda subyacente.

Para obtener más información sobre el uso de la función Ref, consulta [Ref](#) en la Guía del usuario de AWS CloudFormation .

Ejemplos

SimpleTableExample

Ejemplo de SimpleTable

YAML

```
Properties:
  TableName: my-table
  PrimaryKey:
    Name: MyPrimaryKey
    Type: String
  ProvisionedThroughput:
    ReadCapacityUnits: 5
    WriteCapacityUnits: 5
  Tags:
    Department: Engineering
    AppType: Serverless
```

PrimaryKeyObject

El objeto que describe las propiedades de una clave principal.

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
Name: String
```

Type: *String*

Propiedades

Name

Nombre de atributo de la clave principal.

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [AttributeName](#) propiedad del tipo de `AWS::DynamoDB::Table AttributeDefinition` datos.

Notas adicionales: Esta propiedad también se transfiere a la [AttributeName](#) propiedad de un tipo de `AWS::DynamoDB::Table KeySchema` datos.

Type

El tipo de datos de la clave principal.

Valores válidos: `String`, `Number`, `Binary`

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [AttributeType](#) propiedad del tipo de `AWS::DynamoDB::Table AttributeDefinition` datos.

Ejemplos

PrimaryKey

Ejemplo de clave principal

YAML

```
Properties:
  PrimaryKey:
    Name: MyPrimaryKey
```


Type: String

ProvisionedThroughputObject

El objeto describe las propiedades de un rendimiento aprovisionado.

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
ReadCapacityUnits: Integer  
WriteCapacityUnits: Integer
```

Propiedades

ReadCapacityUnits

El número máximo de lecturas altamente coherentes consumidas por segundo antes de que DynamoDB devuelva una excepción `ThrottlingException`.

Tipo: entero

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [ReadCapacityUnits](#) propiedad del tipo de `AWS::DynamoDB::Table ProvisionedThroughput` datos.

WriteCapacityUnits

El número máximo de escrituras consumidas por segundo antes de que DynamoDB devuelva una excepción `ThrottlingException`.

Tipo: entero

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [WriteCapacityUnits](#) propiedad del tipo de `AWS::DynamoDB::Table ProvisionedThroughput` datos.

Ejemplos

ProvisionedThroughput

Ejemplo del rendimiento aprovisionado

YAML

```
Properties:
  ProvisionedThroughput:
    ReadCapacityUnits: 5
    WriteCapacityUnits: 5
```

AWS::Serverless::StateMachine

Crea una máquina de AWS Step Functions estados, que puede utilizar para organizar funciones de AWS Lambda eventos y otros AWS recursos para crear flujos de trabajo complejos y sólidos.

Para obtener más información acerca de Step Functions, consulta la [AWS Step Functions Guía para desarrolladores de](#).

Note

Cuando lo despliega AWS CloudFormation, AWS SAM transforma sus AWS SAM recursos en CloudFormation recursos. Para obtener más información, consulte [Generado CloudFormation recursos para AWS SAM](#).

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
Type: AWS::Serverless::StateMachine
Properties:
  AutoPublishAlias: String
  UseAliasAsEventTarget: Boolean
  Definition: Map
  DefinitionSubstitutions: Map
  DefinitionUri: String | S3Location
```

DeploymentPreference: [DeploymentPreference](#)
Events: [EventSource](#)
Logging: [LoggingConfiguration](#)
Name: *String*
PermissionsBoundary: *String*
Policies: *String | List | Map*
PropagateTags: *Boolean*
RolePath: *String*
Role: *String*
Tags: *Map*
Tracing: [TracingConfiguration](#)
Type: *String*

Propiedades

AutoPublishAlias

El nombre del alias de la máquina de estado. Para obtener más información sobre el uso de los alias de máquinas de estado de Step Functions, consulta [Gestión de implementaciones continuas con versiones y alias](#) en la Guía para desarrolladores de AWS Step Functions .

DeploymentPreferenceUtilícelo para configurar las preferencias de implementación de su alias. Si no lo especificasDeploymentPreference, AWS SAM configurará el tráfico para que cambie a la versión más reciente de la máquina de estados de una sola vez.

AWS SAM establece las versiones DeletionPolicy y en UpdateReplacePolicy de forma Retain predeterminada. Las versiones anteriores no se eliminarán automáticamente.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Name](#) propiedad de un AWS::StepFunctions::StateMachineAlias recurso.

UseAliasAsEventTarget

Indique si quiere pasar o no el alias, creado mediante la propiedad AutoPublishAlias, al destino del origen de eventos definido con [Eventos](#).

Especifique True si quiere utilizar el alias como destino de los eventos.

Tipo: Booleano

Obligatorio: no

Valor predeterminado: `False`

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

Definition

La definición de la máquina de estados es un objeto, donde el formato del objeto coincide con el formato del archivo de AWS SAM plantilla, por ejemplo, JSON o YAML. Las definiciones de máquina de estado cumplen con el [Amazon States Language](#).

Para ver un ejemplo de una definición de máquina de estados en línea, consulta [Ejemplos](#).

Debe proporcionar un `Definition` o un `DefinitionUri`.

Tipo: mapa

Obligatorio: condicional

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

DefinitionSubstitutions

Un mapa (cadena a cadena) que especifica los mapeos para variables de marcador de posición en la definición de máquina de estado. Esto permite al cliente inyectar valores obtenidos en tiempo de ejecución, por ejemplo, a partir de funciones intrínsecas, en la definición de máquina de estado.

Tipo: mapa

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es similar a la [DefinitionSubstitutions](#) propiedad de un `AWS::StepFunctions::StateMachine` recurso. Si se especifica alguna función intrínseca en una definición de máquina de estados en línea, AWS SAM agrega entradas a esta propiedad para incorporarlas a la definición de máquina de estados.

DefinitionUri

La URI de Amazon Simple Storage Service (Amazon S3) o la ruta de archivo local de la definición de la máquina de estados escrita en [Amazon States Language](#).

Si proporcionas una ruta de archivo local, la plantilla debe seguir el flujo de trabajo que incluye el comando `sam deploy` o `sam package` para transformar la definición correctamente. Para hacerlo, debe utilizar la versión 0.52.0 o posterior de la CLI de AWS SAM .

Debe proporcionar un `Definition` o un `DefinitionUri`.

Tipo: Cadena | [S3Location](#)

Obligatorio: condicional

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [DefinitionS3Location](#) propiedad de un `AWS::StepFunctions::StateMachine` recurso.

DeploymentPreference

Los ajustes que permiten y configuran las implementaciones graduales de máquinas de estado. Para obtener más información sobre las implementaciones graduales de Step Functions, consulta [Gestión de implementaciones continuas con versiones y alias](#) en la Guía para desarrolladores de AWS Step Functions .

Especifica esta propiedad `AutoPublishAlias` antes de configurar. La configuración `DeploymentPreference` se aplicará al alias especificado con `AutoPublishAlias`.

Cuando se especifica `DeploymentPreference`, AWS SAM genera el valor de la `StateMachineVersionArn` subpropiedad automáticamente.

Tipo: [DeploymentPreference](#)

Obligatorio: no

CloudFormation compatibilidad: AWS SAM genera y adjunta el valor de la `StateMachineVersionArn` propiedad `DeploymentPreference` y lo transfiere `DeploymentPreference` a la [DeploymentPreference](#) propiedad de un `AWS::StepFunctions::StateMachineAlias` recurso.

Events

Especifica los eventos que activan esta máquina de estados. Los eventos constan de un tipo y un conjunto de propiedades que dependen del tipo.

Tipo: [EventSource](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

Logging

Define qué eventos del historial de ejecución se registran y dónde se registran.

Tipo: [LoggingConfiguration](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [LoggingConfiguration](#) propiedad de un `AWS::StepFunctions::StateMachine` recurso.

Name

El nombre de la máquina de estado.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [StateMachineName](#) propiedad de un `AWS::StepFunctions::StateMachine` recurso.

PermissionsBoundary

El ARN de un límite de permisos que se utilizará para el rol de ejecución de esta máquina de estados. Esta propiedad solo funciona si el rol se genera para usted.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [PermissionsBoundary](#) propiedad de un `AWS::IAM::Role` recurso.

Policies

Políticas de permisos para esta máquina de estados. Las políticas se añadirán a la función de ejecución predeterminada de la máquina de estados AWS Identity and Access Management (IAM).

Esta propiedad acepta un valor único o una lista de valores. Entre los valores permitidos se incluyen:

- [AWS SAM Plantillas de política de](#).
- La ARN de una [política administrada por AWS](#) o una [política administrada por el cliente](#).
- [El nombre de una política AWS administrada de la siguiente lista](#).
- Una [política de IAM en línea](#) con formato en YAML como un mapa.

 Note

Si especifica la propiedad `Role`, pasa por alto esta propiedad.

Tipo: Cadena | Lista | Mapa

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

PropagateTags

Indique si quiere pasar o no las etiquetas de la propiedad `Tags` a los recursos generados [AWS::Serverless::StateMachine](#). Especifica `True` para propagar las etiquetas en los recursos generados.

Tipo: Booleano

Obligatorio: no

Valor predeterminado: `False`

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

Role

El ARN de un rol de IAM que se utilizará como rol de ejecución de esta máquina de estados.

Tipo: cadena

Obligatorio: condicional

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [RoleArn](#) propiedad de un `AWS::StepFunctions::StateMachine` recurso.

RolePath

La ruta al rol de ejecución IAM de la máquina de estados.

Utilice esta propiedad cuando se genere el rol para usted. No la utilice si el rol se especifica con la propiedad `Role`.

Tipo: cadena

Obligatorio: condicional

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Path](#) propiedad de un `AWS::IAM::Role` recurso.

Tags

Una asignación de cadena a cadena que especifica las etiquetas que se agregan a la máquina de estado y la función de ejecución correspondiente. Para obtener información sobre las claves y los valores válidos de las etiquetas, consulta la propiedad [Etiquetas](#) de un recurso [AWS::StepFunctions::StateMachine](#).

Tipo: mapa

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es similar a la [Tags](#) propiedad de un `AWS::StepFunctions::StateMachine` recurso. AWS SAM agrega automáticamente una `stateMachine:createdBy:SAM` etiqueta a este recurso y al rol predeterminado que se genera para él.

Tracing

Selecciona si AWS X-Ray está habilitada o no para la máquina de estados. Para obtener más información sobre el uso X-Ray con Step Functions, consulte [AWS X-Ray Step Functions](#) en la Guía para AWS Step Functions desarrolladores.

Tipo: [TracingConfiguration](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [TracingConfiguration](#) propiedad de un `AWS::StepFunctions::StateMachine` recurso.

Type

El tipo de máquina de estado.

Valores válidos: STANDARD o EXPRESS

Tipo: cadena

Obligatorio: no

Valor predeterminado: STANDARD

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [StateMachineType](#) propiedad de un `AWS::StepFunctions::StateMachine` recurso.

valores devueltos

Ref.

Al proporcionar el ID lógico de este recurso a la función intrínseca Ref, esta devuelve el nombre de recurso de Amazon (ARN) del recurso `AWS::StepFunctions::StateMachine` subyacente.

Para obtener más información sobre el uso de la función Ref, consulta [Ref](#) en la Guía del usuario de AWS CloudFormation .

Ventilador: GetAtt

`Fn::GetAtt` devuelve un valor para un atributo especificado de este tipo. A continuación se indican los atributos disponibles y los valores devueltos de muestra.

Para obtener más información acerca de cómo usar `Fn::GetAtt`, consulta [Fn::GetAtt](#) en la Guía del usuario de AWS CloudFormation .

Name

Devuelve el nombre de la máquina de estados, como `HelloWorld-StateMachine`.

Ejemplos

Archivo de definición de máquina de estado

A continuación, se muestra un ejemplo de una definición de máquina de estados en línea que permite que una función de Lambda invoque una máquina de estados en línea. Tenga en cuenta que en este ejemplo se espera que la propiedad `Role` configure la política adecuada para permitir la invocación. El archivo `my_state_machine.asl.json` debe estar escrito en [Amazon States Language](#).

En este ejemplo, las `DefinitionSubstitution` entradas permiten que la máquina de estados incluya los recursos declarados en el archivo de AWS SAM plantilla.

YAML

```
MySampleStateMachine:
  Type: AWS::Serverless::StateMachine
  Properties:
    DefinitionUri: statemachine/my_state_machine.asl.json
    Role: arn:aws:iam::123456123456:role/service-role/my-sample-role
    Tracing:
      Enabled: true
    DefinitionSubstitutions:
      MyFunctionArn: !GetAtt MyFunction.Arn
      MyDDBTable: !Ref TransactionTable
```

Definición de máquina de estados en línea

A continuación, se muestra un ejemplo de una definición de máquinas de estado insertadas.

En este ejemplo, el archivo de AWS SAM plantilla está escrito en YAML, por lo que la definición de la máquina de estados también está en YAML. Para declarar una definición de máquina de estados integrada en JSON, escribe el archivo de AWS SAM plantilla en JSON.

YAML

```
MySampleStateMachine:
  Type: AWS::Serverless::StateMachine
  Properties:
    Definition:
      StartAt: MyLambdaState
    States:
```

```

MyLambdaState:
  Type: Task
  Resource: arn:aws:lambda:us-east-1:123456123456:function:my-sample-lambda-app
  End: true
Role: arn:aws:iam::123456123456:role/service-role/my-sample-role
Tracing:
  Enabled: true

```

EventSource

El objeto que describe la fuente de los eventos que activan la máquina de estados. Cada evento consta de un tipo y un conjunto de propiedades que dependen de ese tipo. Para obtener información acerca de las propiedades de cada fuente de eventos, consulta el tema secundario correspondiente a ese tipo.

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```

Properties: Schedule | ScheduleV2 | CloudWatchEvent | EventBridgeRule | Api
Type: String

```

Propiedades

Properties

Un objeto que describe las propiedades de este mapeo de eventos. El conjunto de propiedades debe ajustarse al Type definido.

Tipo: [Schedule](#) | [ScheduleV2](#) | [CloudWatchEvent](#) | [EventBridgeRule](#) | [Api](#)

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

Type

Tipo de evento.

Valores válidos: `Api`, `Schedule`, `ScheduleV2`, `CloudWatchEvent`, `EventBridgeRule`

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

Ejemplos

API

A continuación, se muestra un ejemplo de un evento tipo API.

YAML

```
ApiEvent:
  Type: Api
  Properties:
    Method: get
    Path: /group/{user}
    RestApiId:
      Ref: MyApi
```

Api

El objeto que describe un tipo de fuente de evento de `Api`. Si se define un recurso [AWS::Serverless::API](#), los valores de ruta y método deben corresponder a una operación de la definición de OpenAPI de la API.

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
Auth: ApiStateMachineAuth
Method: String
Path: String
RestApiId: String
```

[UnescapeMappingTemplate](#): *Boolean*

Propiedades

Auth

La configuración de autorización de esta API, ruta y método.

Usa esta propiedad para anular la configuración `DefaultAuthorizer` de la API para una ruta individual, si no se especifica `DefaultAuthorizer`, o para anular la configuración de `ApiKeyRequired` predeterminada.

Tipo: [ApiStateMachineAuth](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

Method

El método HTTP para el que se invoca esta función.

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

Path

La ruta URI para la que se invoca esta función. El valor debe empezar por `/`.

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

RestApiId

El identificador de un recurso `RestApi`, que debe contener una operación con la ruta y el método dados. Normalmente, se configura para hacer referencia a un recurso [AWS::Serverless::API](#) definido en esta plantilla.

Si no define esta propiedad, AWS SAM crea un [AWS::Serverless::API](#) recurso predeterminado a partir de un OpenApi documento generado. Ese recurso contiene una unión de todas las rutas y métodos definidos por eventos Api de la misma plantilla que no especifican un RestApiId.

Esta propiedad no puede hacer referencia a un recurso [AWS::Serverless::API](#) definido en otra plantilla.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

UnescapeMappingTemplate

Evita las comillas simples, sustituyendo \ ' por ' , en la entrada que se pasa a la máquina de estados. Haga esto cuando la entrada contenga comillas simples.

Note

Si se establece en False y la entrada contiene comillas simples, se producirá un error.

Tipo: Booleano

Obligatorio: no

Valor predeterminado: False

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

Ejemplos

ApiEvent

A continuación, se muestra un ejemplo de un evento tipo Api.

YAML

```
Events:
```

```
ApiEvent:
  Type: Api
  Properties:
    Path: /path
    Method: get
```

ApiStateMachineAuth

Configura la autorización a nivel de evento para una API, una ruta y un método específicos.

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
ApiKeyRequired: Boolean
AuthorizationScopes: List
Authorizer: String
ResourcePolicy: ResourcePolicyStatement
```

Propiedades

ApiKeyRequired

Requiere una clave de API para esta API, ruta y método.

Tipo: Booleano

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

AuthorizationScopes

Los ámbitos de autorización que se aplicarán a esta API, ruta y método.

Los ámbitos que especifica anularán los ámbitos aplicados por la propiedad de `DefaultAuthorizer` si los ha especificado.

Tipo: lista

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

Authorizer

El Authorizer para una máquina de estados específica.

Si especificaste un autorizador global para la API y quieres que esta máquina de estados sea pública, anula el autorizador global configurando Authorizer en NONE.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

ResourcePolicy

Configura la política de recursos para esta API y esta ruta.

Tipo: [ResourcePolicyStatement](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

Ejemplos

StateMachine-Autenticación

El siguiente ejemplo especifica la autorización a nivel de máquina de estado.

YAML

```
Auth:
  ApiKeyRequired: true
  Authorizer: NONE
```


ResourcePolicyStatement

Configura una política de recursos para todos los métodos y rutas de una API. Para obtener más información sobre las políticas de recursos, consulta [Controlar el acceso a una API con las políticas de recursos de API Gateway](#) en la Guía para desarrolladores de API Gateway.

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
AwsAccountBlacklist: List
AwsAccountWhitelist: List
CustomStatements: List
IntrinsicVpcBlacklist: List
IntrinsicVpcWhitelist: List
IntrinsicVpceBlacklist: List
IntrinsicVpceWhitelist: List
IpRangeBlacklist: List
IpRangeWhitelist: List
SourceVpcBlacklist: List
SourceVpcWhitelist: List
```

Propiedades

AwsAccountBlacklist

Las AWS cuentas que se van a bloquear.

Tipo: lista de cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

AwsAccountWhitelist

Las AWS cuentas que se van a permitir. Para obtener un ejemplo del uso de esta propiedad, consulta la sección Ejemplos al final de esta página.

Tipo: lista de cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

CustomStatements

Una lista de instrucciones de política de recursos personalizadas para aplicarlas a esta API. Para obtener un ejemplo del uso de esta propiedad, consulta la sección Ejemplos al final de esta página.

Tipo: lista

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

IntrinsicVpcBlacklist

La lista de nubes privadas virtuales (VPCs) que se van a bloquear, en la que cada VPC se especifica como referencia, por ejemplo, una [referencia dinámica](#) o la función Ref [intrínseca](#). Para obtener un ejemplo del uso de esta propiedad, consulta la sección Ejemplos al final de esta página.

Tipo: lista

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

IntrinsicVpcWhitelist

La lista de VPCs permitidos, donde cada VPC se especifica como una referencia, como una [referencia dinámica](#) o la función Ref [intrínseca](#).

Tipo: lista

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

IntrinsicVpceBlacklist

La lista de puntos de conexión de VPC que se van a bloquear, donde cada punto de conexión de VPC se especifica como referencia, por ejemplo, una [referencia dinámica](#) o la [función intrínseca Ref](#).

Tipo: lista

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

IntrinsicVpceWhitelist

La lista de nubes privadas virtuales (VPC) permitidas, donde cada punto de conexión de VPC se especifica como referencia, por ejemplo, una [referencia dinámica](#) o la [función intrínseca Ref](#). Para obtener un ejemplo del uso de esta propiedad, consulta la sección Ejemplos al final de esta página.

Tipo: lista

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

IpRangeBlacklist

Las direcciones IP o los rangos de direcciones que se van a bloquear. Para obtener un ejemplo del uso de esta propiedad, consulta la sección Ejemplos al final de esta página.

Tipo: lista

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

IpRangeWhitelist

Las direcciones IP o los rangos de direcciones que se van a permitir.

Tipo: lista

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

SourceVpcBlacklist

La VPC de origen o los puntos de conexión de VPC que se van a bloquear. Los nombres de la VPC de origen deben empezar por "vpc-" y los nombres de los puntos de conexión de VPC de origen deben empezar por "vpce-". Para obtener un ejemplo del uso de esta propiedad, consulta la sección Ejemplos al final de esta página.

Tipo: lista

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

SourceVpcWhitelist

La VPC de origen o los puntos de conexión de VPC que se van a permitir. Los nombres de la VPC de origen deben empezar por "vpc-" y los nombres de los puntos de conexión de VPC de origen deben empezar por "vpce-".

Tipo: lista

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

Ejemplos

Ejemplo de política de recursos

El siguiente ejemplo bloquea dos direcciones IP y una VPC de origen y permite una AWS cuenta.

YAML

```
Auth:
  ResourcePolicy:
    CustomStatements: [{
      "Effect": "Allow",
```

```

        "Principal": "*",
        "Action": "execute-api:Invoke",
        "Resource": "execute-api:/Prod/GET/pets",
        "Condition": {
            "IpAddress": {
                "aws:SourceIp": "1.2.3.4"
            }
        }
    }
}]]

IpRangeBlacklist:
  - "10.20.30.40"
  - "1.2.3.4"
SourceVpcBlacklist:
  - "vpce-1a2b3c4d"
AwsAccountWhitelist:
  - "111122223333"
IntrinsicVpcBlacklist:
  - "{{resolve:ssm:SomeVPCReference:1}}"
  - !Ref MyVPC
IntrinsicVpceWhitelist:
  - "{{resolve:ssm:SomeVPCEReference:1}}"
  - !Ref MyVPCE

```

CloudWatchEvent

El objeto que describe un tipo de fuente de evento CloudWatchEvent.

AWS Serverless Application Model (AWS SAM) genera un [AWS::Events::Rule](#) recurso cuando se establece este tipo de evento.

Nota importante: [EventBridgeRule](#) es el tipo de fuente de eventos preferido para usar, en lugar de CloudWatchEvent. EventBridgeRule CloudWatchEvent usa el mismo servicio, API y CloudFormation recursos subyacentes. Sin embargo, solo AWS SAM agregará soporte para nuevas funciones para EventBridgeRule.

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
EventBusName: String
```

```
Input: String  
InputPath: String  
Pattern: EventPattern
```

Propiedades

EventBusName

El bus de eventos que se asociará a esta regla. Si omite esta propiedad, AWS SAM utiliza el bus de eventos predeterminado.

Tipo: cadena

Obligatorio: no

Predeterminado: bus de eventos predeterminado

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [EventBusName](#) propiedad de un `AWS::Events::Rule` recurso.

Input

Texto JSON válido transmitido al destino. Si utiliza esta propiedad, no se transmitirá nada del propio texto del evento al destino.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Input](#) propiedad de un `AWS::Events::Rule` Target recurso.

InputPath

Cuando no desee pasar todo el evento coincidente al destino, utilice la propiedad `InputPath` para describir qué parte del evento pasar.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [InputPath](#) propiedad de un `AWS::Events::Rule` Target recurso.

Pattern

Describe qué eventos se dirigen al destino especificado. Para obtener más información, consulte [Eventos y patrones de eventos EventBridge en](#) la Guía del EventBridge usuario de Amazon.

Tipo: [EventPattern](#)

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [EventPattern](#) propiedad de un `AWS::Events::Rule` recurso.

Ejemplos

CloudWatchEvent

A continuación, se muestra un ejemplo de un tipo de fuente de evento `CloudWatchEvent`.

YAML

```
CWEvent:
  Type: CloudWatchEvent
  Properties:
    Input: '{"Key": "Value"}'
    Pattern:
      detail:
        state:
          - running
```

EventBridgeRule

El objeto que describe un tipo de fuente de `EventBridgeRule` eventos, que establece tu máquina de estados como el destino de una EventBridge regla de Amazon. Para obtener más información, consulta [¿Qué es Amazon EventBridge?](#) en la Guía del EventBridge usuario de Amazon.

AWS SAM genera un [AWS::Events::Rule](#) recurso cuando se establece este tipo de evento.

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
DeadLetterConfig: DeadLetterConfig  
EventBusName: String  
Input: String  
InputPath: String  
InputTransformer: InputTransformer  
Pattern: EventPattern  
RetryPolicy: RetryPolicy  
RuleName: String  
State: String  
Target: Target
```

Propiedades

DeadLetterConfig

Configure la cola del Amazon Simple Queue Service (Amazon SQS), desde la que EventBridge se envían los eventos tras una invocación de destino fallida. La invocación puede fallar, por ejemplo, cuando se envía un evento a una función de eventos de Lambda que no existe o EventBridge cuando no hay permisos suficientes para invocar la función de Lambda. Para obtener más información, consulta la [política de reintentos de eventos y el uso de colas con letra muerta en la Guía del usuario](#) de Amazon. EventBridge

Tipo: [DeadLetterConfig](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es similar a la [DeadLetterConfig](#) propiedad del tipo de datos. `AWS::Events::Rule` Target La AWS SAM versión de esta propiedad incluye subpropiedades adicionales, por si AWS SAM quiere crear automáticamente una cola de texto sin procesar.

EventBusName

El bus de eventos que se asociará a esta regla. Si omite esta propiedad, AWS SAM utiliza el bus de eventos predeterminado.

Tipo: cadena

Obligatorio: no

Predeterminado: bus de eventos predeterminado

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [EventBusName](#) propiedad de un `AWS::Events::Rule` recurso.

Input

Texto JSON válido transmitido al destino. Si utiliza esta propiedad, no se transmitirá nada del propio texto del evento al destino.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Input](#) propiedad de un `AWS::Events::Rule Target` recurso.

InputPath

Cuando no desee pasar todo el evento coincidente al destino, utilice la propiedad `InputPath` para describir qué parte del evento pasar.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [InputPath](#) propiedad de un `AWS::Events::Rule Target` recurso.

InputTransformer

Configuración que permite proporcionar una entrada personalizada a un destino en función de determinados datos de eventos. Puedes extraer uno o más pares clave-valor del evento y, a continuación, utilizar dichos datos para enviar una entrada personalizada al destino. Para obtener más información, consulta la [transformación EventBridge de entradas de Amazon](#) en la Guía del EventBridge usuario de Amazon.

Tipo: [InputTransformer](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [InputTransformer](#) propiedad de un tipo de `AWS::Events::Rule Target` datos.

Pattern

Describe qué eventos se dirigen al destino especificado. Para obtener más información, consulte [Eventos y patrones de eventos EventBridge en](#) la Guía del EventBridge usuario de Amazon.

Tipo: [EventPattern](#)

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [EventPattern](#) propiedad de un `AWS::Events::Rule` recurso.

RetryPolicy

Objeto de la `RetryPolicy` que incluye información acerca de la configuración de la política de reintentos. Para obtener más información, consulta la [política de reintentos de eventos y el uso de colas con letra muerta en la Guía del usuario](#) de Amazon. EventBridge

Tipo: [RetryPolicy](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [RetryPolicy](#) propiedad del tipo de datos. `AWS::Events::Rule Target`

RuleName

El nombre de la regla.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Name](#) propiedad de un `AWS::Events::Rule` recurso.

State

El estado de la regla.

Valores válidos: [DISABLED | ENABLED]

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [State](#) propiedad de un `AWS::Events::Rule` recurso.

Target

El AWS recurso que se EventBridge invoca cuando se activa una regla. Puedes utilizar esta propiedad para especificar el ID lógico del objetivo. Si no se especifica esta propiedad, AWS SAM genera el ID lógico del objetivo.

Tipo: [destino](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es similar a la [Targets](#) propiedad de un `AWS::Events::Rule` recurso. La versión AWS SAM de esta propiedad solo permite especificar el identificador lógico de un único destino.

Ejemplos

EventBridgeRule

A continuación, se muestra un ejemplo de un tipo de fuente de evento EventBridgeRule.

YAML

```
EBRule:
  Type: EventBridgeRule
  Properties:
    Input: '{"Key": "Value"}'
    Pattern:
      detail:
        state:
          - terminated
```

DeadLetterConfig

El objeto utilizado para especificar la cola del Amazon Simple Queue Service (Amazon SQS) desde la que EventBridge se envían los eventos tras una invocación de destino fallida. La invocación puede fallar, por ejemplo, cuando se envía un evento a una máquina de estado que no existe o cuando no

hay permisos suficientes para invocar la máquina de estado. Para obtener más información, consulta la [política de reintentos de eventos y el uso de colas con letra muerta en la Guía del usuario](#) de Amazon. EventBridge

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
Arn: String  
QueueLogicalId: String  
Type: String
```

Propiedades

Arn

El nombre de recurso de Amazon (ARN) de la cola SQS de Amazon especificada como destino de la cola de mensajes fallidos.

Note

Especifique la propiedad Type o la propiedad Arn, pero no ambas.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Arn](#) propiedad del tipo de `AWS::Events::Rule DeadLetterConfig` datos.

QueueLogicalId

TypeSe especifica el nombre personalizado de la cola de cartas muertas que la AWS SAM crea.

Note

Si no se especifica la propiedad Type, entonces esta propiedad se pasa por alto.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

Type

Tipo de cola. Cuando se establece esta propiedad, crea AWS SAM automáticamente una cola sin efecto y adjunta la [política basada en los recursos](#) necesaria para conceder permiso al recurso de regla para enviar eventos a la cola.

Note

Especifica la propiedad Type o la propiedad Arn, pero no ambas.

Valores válidos: SQS

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva y no tiene un equivalente. AWS SAM CloudFormation

Ejemplos

DeadLetterConfig

DeadLetterConfig

YAML

```
DeadLetterConfig:
  Type: SQS
  QueueLogicalId: MyDLQ
```

Target

Configura el AWS recurso que se EventBridge invoca cuando se activa una regla.

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
Id: String
```

Propiedades

Id

El ID lógico del destino.

El valor de Id puede contener caracteres alfanuméricos, puntos (.), guiones (-) y guiones bajos (_).

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Id](#) propiedad del tipo de `AWS::Events::Rule` Target datos.

Ejemplos

Target

YAML

```
EBRule:
  Type: EventBridgeRule
  Properties:
    Target:
      Id: MyTarget
```

Schedule

El objeto que describe un tipo de fuente de Schedule eventos, que establece su máquina de estados como el objetivo de una EventBridge regla que se activa según una programación. Para

obtener más información, consulta [¿Qué es Amazon EventBridge?](#) en la Guía del EventBridge usuario de Amazon.

AWS Serverless Application Model (AWS SAM) genera un [AWS::Events::Rule](#) recurso cuando se establece este tipo de evento.

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
DeadLetterConfig: DeadLetterConfig
Description: String
Enabled: Boolean
Input: String
Name: String
RetryPolicy: RetryPolicy
RoleArn: String
Schedule: String
State: String
Target: Target
```

Propiedades

DeadLetterConfig

Configure la cola del Amazon Simple Queue Service (Amazon SQS), desde la que EventBridge se envían los eventos tras una invocación de destino fallida. La invocación puede fallar, por ejemplo, cuando se envía un evento a una función de eventos de Lambda que no existe o EventBridge cuando no hay permisos suficientes para invocar la función de Lambda. Para obtener más información, consulta la [política de reintentos de eventos y el uso de colas con letra muerta en la Guía del usuario](#) de Amazon. EventBridge

Tipo: [DeadLetterConfig](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es similar a la [DeadLetterConfig](#) propiedad del tipo de datos. `AWS::Events::Rule Target` La AWS SAM versión de esta propiedad

incluye subpropiedades adicionales, por si AWS SAM quiere crear automáticamente una cola de texto sin procesar.

Description

Una descripción del conjunto de reglas.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [Description](#) propiedad de un recurso. `AWS::Events::Rule`

Enabled

Indica si la regla está habilitada.

Para deshabilitar la regla, defina esta propiedad en `false`.

Note

Especifique la propiedad `Enabled` o `State`, pero no ambas.

Tipo: Booleano

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es similar a la [State](#) propiedad de un `AWS::Events::Rule` recurso. Si esta propiedad se establece en, `true` entonces, se AWS SAM aprueba; de lo contrario `ENABLED`, se aprueba `DISABLED`.

Input

Texto JSON válido transmitido al destino. Si utiliza esta propiedad, no se transmitirá nada del propio texto del evento al destino.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Input](#) propiedad de un `AWS::Events::Rule` `Target` recurso.

Name

El nombre de la regla. Si no especificas un nombre, CloudFormation genera un identificador físico único y lo utiliza como nombre de la regla.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [Name](#) propiedad de un `AWS::Events::Rule` recurso.

RetryPolicy

Objeto de la `RetryPolicy` que incluye información acerca de la configuración de la política de reintentos. Para obtener más información, consulta la [política de reintentos de eventos y el uso de colas con letra muerta en la Guía del usuario](#) de Amazon. EventBridge

Tipo: [RetryPolicy](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [RetryPolicy](#) propiedad del tipo de datos. `AWS::Events::Rule Target`

RoleArn

El ARN de la función de IAM que EventBridge Scheduler utilizará para el destino cuando se invoque la programación.

Tipo: [RoleArn](#)

Requerido: no. Si no se proporciona, se creará y utilizará un nuevo rol.

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [RoleArn](#) propiedad del tipo de `AWS::Scheduler::Schedule Target` datos.

Schedule

La expresión de programación que determina cuándo y con qué frecuencia se ejecuta la regla. Para obtener más información, consulta [Expresiones de programación para reglas](#).

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [ScheduleExpression](#) propiedad de un `AWS::Events::Rule` recurso.

State

El estado de la regla.

Valores aceptados DISABLED | ENABLED:

Note

Especifique la propiedad Enabled o State, pero no ambas.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [State](#) propiedad de un `AWS::Events::Rule` recurso.

Target

El AWS recurso que se EventBridge invoca cuando se activa una regla. Puedes utilizar esta propiedad para especificar el ID lógico del objetivo. Si no se especifica esta propiedad, AWS SAM genera el ID lógico del objetivo.

Tipo: [destino](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es similar a la [Targets](#) propiedad de un `AWS::Events::Rule` recurso. La AWS SAM versión de esta propiedad solo permite especificar el identificador lógico de un único destino.

Ejemplos

CloudWatch Programe un evento

CloudWatch Ejemplo de programar un evento

YAML

```
CWSchedule:
  Type: Schedule
  Properties:
    Schedule: 'rate(1 minute)'
    Name: TestSchedule
    Description: test schedule
    Enabled: false
```

DeadLetterConfig

El objeto utilizado para especificar la cola del Amazon Simple Queue Service (Amazon SQS) desde la que EventBridge se envían los eventos tras una invocación de destino fallida. La invocación puede fallar, por ejemplo, cuando se envía un evento a una máquina de estado que no existe o cuando no hay permisos suficientes para invocar la máquina de estado. Para obtener más información, consulta la [política de reintentos de eventos y el uso de colas con letra muerta en la Guía del usuario](#) de Amazon. EventBridge

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
Arn: String
QueueLogicalId: String
Type: String
```

Propiedades

Arn

El nombre de recurso de Amazon (ARN) de la cola SQS de Amazon especificada como destino de la cola de mensajes fallidos.

Note

Especifique la propiedad `Type` o la propiedad `Arn`, pero no ambas.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Arn](#) propiedad del tipo de `AWS::Events::Rule` DeadLetterConfig datos.

QueueLogicalId

TypeSe especifica el nombre personalizado de la cola de cartas muertas que la AWS SAM crea.

 Note

Si no se especifica la propiedad Type, entonces esta propiedad se pasa por alto.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

Type

Tipo de cola. Cuando se establece esta propiedad, crea AWS SAM automáticamente una cola sin efecto y adjunta la [política basada en los recursos](#) necesaria para conceder permiso al recurso de regla para enviar eventos a la cola.

 Note

Especifica la propiedad Type o la propiedad Arn, pero no ambas.

Valores válidos: SQS

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva y no tiene un equivalente. AWS SAM CloudFormation

Ejemplos

DeadLetterConfig

DeadLetterConfig

YAML

```
DeadLetterConfig:  
  Type: SQS  
  QueueLogicalId: MyDLQ
```

Target

Configura el AWS recurso que se EventBridge invoca cuando se activa una regla.

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
Id: String
```

Propiedades

Id

El ID lógico del destino.

El valor de Id puede contener caracteres alfanuméricos, puntos (.), guiones (-) y guiones bajos (_).

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Id](#) propiedad del tipo de `AWS::Events::Rule` Target datos.

Ejemplos

Target

YAML

```
EBRule:
  Type: Schedule
  Properties:
    Target:
      Id: MyTarget
```

ScheduleV2

El objeto que describe un tipo de fuente de `ScheduleV2` eventos, que establece su máquina de estados como el objetivo de un evento de Amazon EventBridge Scheduler que se desencadena de forma programada. Para obtener más información, consulte [¿Qué es Amazon EventBridge Scheduler?](#) en la Guía del usuario de EventBridge Scheduler.

AWS Serverless Application Model (AWS SAM) genera un [AWS::Scheduler::Schedule](#) recurso cuando se establece este tipo de evento.

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
DeadLetterConfig: DeadLetterConfig
Description: String
EndDate: String
FlexibleTimeWindow: FlexibleTimeWindow
GroupName: String
Input: String
KmsKeyArn: String
Name: String
OmitName: Boolean
PermissionsBoundary: String
RetryPolicy: RetryPolicy
RoleArn: String
ScheduleExpression: String
ScheduleExpressionTimezone: String
```

```
StartDate: String  
State: String
```

Propiedades

DeadLetterConfig

Configure la cola del Amazon Simple Queue Service (Amazon SQS), desde la que EventBridge se envían los eventos tras una invocación de destino fallida. La invocación puede fallar, por ejemplo, cuando se envía un evento a una función de eventos de Lambda que no existe o EventBridge cuando no hay permisos suficientes para invocar la función de Lambda. Para obtener más información, consulte [Configuración de una cola de letra muerta para Scheduler en la Guía del usuario de Scheduler EventBridge](#). EventBridge

Tipo: [DeadLetterConfig](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es similar a la [DeadLetterConfig](#) propiedad del tipo de datos. `AWS::Scheduler::ScheduleTarget` La AWS SAM versión de esta propiedad incluye subpropiedades adicionales, por si AWS SAM quiere crear automáticamente una cola de texto sin procesar.

Description

Una descripción del programa.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [Description](#) propiedad de un recurso. `AWS::Scheduler::Schedule`

EndDate

La fecha, en UTC, antes de la cual la programación puede invocar su destino. Según la expresión de recurrencia de la programación, las invocaciones se pueden detener en la EndDate que especifica o antes de ella.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [EndDate](#) propiedad de un `AWS::Scheduler::Schedule` recurso.

FlexibleTimeWindow

Permite configurar una ventana en la cual se puede invocar una programación.

Tipo: [FlexibleTimeWindow](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [FlexibleTimeWindow](#) propiedad de un `AWS::Scheduler::Schedule` recurso.

GroupName

El nombre de grupo de programación asociado a esta programación. Si no está definido, se utiliza el grupo predeterminado.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [GroupName](#) propiedad de un `AWS::Scheduler::Schedule` recurso.

Input

Texto JSON válido transmitido al destino. Si utiliza esta propiedad, no se transmitirá nada del propio texto del evento al destino.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Input](#) propiedad de un `AWS::Scheduler::Schedule Target` recurso.

KmsKeyArn

El ARN de una clave de KMS que se utiliza para cifrar datos personalizados.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [KmsKeyArn](#) propiedad de un `AWS::Scheduler::Schedule` recurso.

Name

El nombre de la programación. Si no especifica un nombre, AWS SAM genera un nombre en el formato *StateMachine-Logical-IDEvent-Source-Name* y usa ese ID como nombre del programa.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [Name](#) propiedad de un `AWS::Scheduler::Schedule` recurso.

OmitName

De forma predeterminada, AWS SAM genera y usa un nombre de programación con el formato de *<State-machine-logical-ID><event-source-name>*. Establezca esta propiedad `true` para CloudFormation generar un identificador físico único y utilícelo en su lugar para el nombre del programa.

Tipo: Booleano

Obligatorio: no

Valor predeterminado: `false`

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

PermissionsBoundary

El ARN de la política que se utilizará para definir el límite de los permisos para el rol.

Note

Si `PermissionsBoundary` está definido, AWS SAM aplicará los mismos límites a la función de IAM objetivo del programador.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [PermissionsBoundary](#) propiedad de un `AWS::IAM::Role` recurso.

RetryPolicy

Objeto de la `RetryPolicy` que incluye información acerca de la configuración de la política de reintentos.

Tipo: [RetryPolicy](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [RetryPolicy](#) propiedad del tipo de `AWS::Scheduler::Schedule Target` datos.

RoleArn

El ARN de la función de IAM que EventBridge Scheduler utilizará para el destino cuando se invoque la programación.

Tipo: [RoleArn](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [RoleArn](#) propiedad del tipo de `AWS::Scheduler::Schedule Target` datos.

ScheduleExpression

La expresión de programación que determina cuándo y con qué frecuencia se ejecuta la programación.

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [ScheduleExpression](#) propiedad de un `AWS::Scheduler::Schedule` recurso.

ScheduleExpressionTimezone

La zona horaria en la que se evalúa la expresión de programación.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [ScheduleExpressionTimezone](#) propiedad de un `AWS::Scheduler::Schedule` recurso.

StartDate

La fecha, en UTC, después de la cual la programación puede invocar un objetivo. Según la expresión de recurrencia de la programación, las invocaciones se pueden realizar en la `StartDate` que especifica o después de ella.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [StartDate](#) propiedad de un `AWS::Scheduler::Schedule` recurso.

State

Estado del programa.

Valores aceptados: DISABLED | ENABLED

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [State](#) propiedad de un `AWS::Scheduler::Schedule` recurso.

Ejemplos

Ejemplo básico de definición de un recurso de `ScheduleV2`

```
StateMachine:
  Type: AWS::Serverless::StateMachine
  Properties:
    Name: MyStateMachine
    Events:
      ScheduleEvent:
        Type: ScheduleV2
        Properties:
          ScheduleExpression: "rate(1 minute)"
      ComplexScheduleEvent:
        Type: ScheduleV2
```

```
Properties:
  ScheduleExpression: rate(1 minute)
  FlexibleTimeWindow:
    Mode: FLEXIBLE
    MaximumWindowInMinutes: 5
  StartDate: '2022-12-28T12:00:00.000Z'
  EndDate: '2023-01-28T12:00:00.000Z'
  ScheduleExpressionTimezone: UTC
  RetryPolicy:
    MaximumRetryAttempts: 5
    MaximumEventAgeInSeconds: 300
  DeadLetterConfig:
    Type: SQS
DefinitionUri:
  Bucket: sam-sam-s3-demo-bucket
  Key: my-state-machine.asl.json
  Version: 3
Policies:
  - LambdaInvokePolicy:
    FunctionName: !Ref MyFunction
```

AWS::Serverless::WebFunction

Crea una función AWS Lambda web, que atiende las solicitudes HTTP a través de un punto final gestionado. A diferencia de una función de eventos Lambda, que se ejecuta en respuesta a invocaciones y eventos, una función web ejecuta su servidor web y recibe solicitudes HTTP directamente. A partir de una sola `AWS::Serverless::WebFunction`, AWS SAM genera la función web, una revisión inmutable que contiene el código y la configuración, y un punto final que dirige el tráfico a una o más revisiones.

Note

Al realizar la implementación en AWS CloudFormation, AWS SAM transforma sus AWS SAM recursos en CloudFormation recursos. Para obtener más información, consulte [Generado CloudFormation recursos para AWS SAM](#).

Important

El código de la función debe ejecutar un servidor HTTP que escuche el host y el puerto de la variable de `AWS_LAMBDA_HTTP_ENDPOINT` entorno. El tiempo de ejecución no está

configurado `PORT`. De forma predeterminada, el tiempo de ejecución se ejecuta `index.js` en la raíz del paquete de despliegue. Se utiliza [EntryPoint](#) para ejecutar un archivo diferente.

Important

El depósito de Amazon S3 que contiene el paquete de implementación debe cumplir los siguientes requisitos:

- El control de versiones está activado.
- Una política de bucket otorga `lambda.amazonaws.com` los `s3:GetObjectVersion` permisos `s3:GetObject` y. Le recomendamos que defina la política con una `ArnLike` condición `aws:SourceArn` que coincida con las revisiones de la función, por ejemplo `arn:aws:lambda:us-east-1:111122223333:web-function/my-function/revision/*`.
- Si el objeto está cifrado con SSE-KMS, utiliza una clave gestionada por el cliente cuya política de claves permite `lambda.amazonaws.com` realizar `llamada:kms:Decrypt`. La clave AWS gestionada de Amazon S3 (`aws/s3`) no es compatible porque no se puede cambiar su política de claves.

Para ver un ejemplo de política de bucket, consulte [Política de bucket para Lambda](#) en la Guía para AWS Lambda desarrolladores.

`CodeUri` debe hacer referencia a un objeto en Amazon S3. Los `sam package` comandos `sam build` y no admiten las rutas locales de este recurso.

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
Type: AWS::Serverless::WebFunction
Properties:
  AuthType: String
  AutoDeploymentMode: String
```

```
CodeUri: String | CodeUri  
EgressNetworkConnectorArn: String  
EndpointDescription: String  
EndpointName: String  
EndpointType: String  
EntryPoint: String  
EnvironmentVariables: Map  
ExecutionRoleArn: String  
FunctionName: String  
InlineCode: String  
KmsKeyArn: String  
LoggingConfig: Map  
MaxConcurrencyPerEnvironment: Integer  
ReplicaRegions: List | ReplicaRegions  
RevisionDescription: String  
RevisionWeights: RevisionWeights  
Runtime: String  
ScalingConfig: Map  
Tags: Map  
ThrottleConfig: Map  
Timeout: Integer
```

Propiedades

AuthType

Cómo autoriza el punto final las solicitudes.

- **IamAuth**: Lambda autoriza las solicitudes con Signature Version 4 e IAM. Las personas que llaman necesitan el `lambda:InvokeWebFunctionEndpoint` permiso, que se otorga mediante una política basada en los recursos de la función.
- **ApplicationManaged**: el punto final es público. Su función debe autorizar cada solicitud por sí misma.

Valores válidos: `IamAuth`, `ApplicationManaged`

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [AuthType](#) propiedad de un `AWS::Lambda::WebFunctionEndpoint` recurso.

AutoDeploymentMode

Cómo el punto final enruta el tráfico entre las revisiones.

- **LatestRevision**: envía el 100% del tráfico a la revisión que crea esta implementación.
- **Disabled**: enruta el tráfico según [RevisionWeights](#).

Los valores que se pueden utilizar y si **RevisionWeights** son obligatorios dependen de [EndpointType](#):

- **HomeRegion**: utilice cualquiera de los dos valores. Con **Disabled**, **RevisionWeights** es obligatorio.
- **MultiRegionPerRegion**: usar **Disabled**. Si lo omite **RevisionWeights**, el punto final redirige el 100% del tráfico a la revisión creada en la misma pila, de modo que puede crear el punto final en una sola implementación. El tráfico no pasa a las revisiones que crean las implementaciones posteriores.

El valor debe resolverse al AWS SAM transformar la plantilla. Se admite el valor Ref A a un parámetro. No se admiten otras funciones intrínsecas.

Valores válidos: **LatestRevision**, **Disabled**

Tipo: cadena

Obligatorio: no

Valor predeterminado: **LatestRevision**

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

CodeUri

La ubicación en Amazon S3 del paquete de implementación. Especifique un `s3://bucket/key` URI o un objeto con **BucketKey**, y **Version** claves opcionales. Para conocer los requisitos del bucket, consulta la nota que aparece en la parte superior de esta página.

Tipo: Cadena | [CodeUri](#)

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad es similar a la [BuildConfig.CodeConfig.S3Object](#) propiedad de un

`AWS::Lambda::WebFunctionRevision` recurso. AWS SAM convierte un `s3://` URI en `BucketKey`, y `VersionId`.

EgressNetworkConnectorArn

El ARN compatible con la versión de un conector de red que enruta el tráfico saliente de la función a través de la VPC. El ARN debe terminar en un número de versión, por ejemplo.

`arn:aws:lambda:us-east-1:123456789012:network-connector:my-connector:1`

Para usar la última versión de un conector en la misma plantilla, especifique `!GetAtt`

`MyConnector.LatestVersionArn`. Los conectores [ComputeTypes](#) deben ser `WebFunction`.

Si especifica una cadena literal, AWS SAM comprueba que se trate de un ARN de conector de red compatible con la versión. El servicio transmite y valida las funciones intrínsecas. Al cambiar este valor, se crea una nueva revisión.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [ServiceConfig.EgressNetworkConnectorArn](#) propiedad de un `AWS::Lambda::WebFunctionRevision` recurso.

EndpointDescription

Una descripción del punto final.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Description](#) propiedad de un `AWS::Lambda::WebFunctionEndpoint` recurso.

EndpointName

El nombre del punto de enlace. Al cambiar el nombre, se reemplaza el punto final. El cambio `EndpointType` también reemplaza el punto final, así que `EndpointName` cámbielo al mismo tiempo. CloudFormation no se puede reemplazar un recurso con un nombre personalizado sin un nombre nuevo.

Tipo: cadena

Obligatorio: no

Valor predeterminado: prod

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [EndpointName](#) propiedad de un `AWS::Lambda::WebFunctionEndpoint` recurso.

EndpointType

Dónde el punto final atiende el tráfico.

- `HomeRegion`: sirve desde la región de la pila.
- `MultiRegion`: replica la función en las regiones [ReplicaRegions](#) y dirige cada solicitud a la región más cercana.
- `PerRegion`: muestra un punto final independiente en cada región de réplica.

`MultiRegion` y `PerRegion` requieren `AutoDeploymentMode: Disabled`.

Valores válidos: `HomeRegion`, `MultiRegion`, `PerRegion`

Tipo: cadena

Obligatorio: no

Valor predeterminado: `HomeRegion`

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [EndpointType](#) propiedad de un `AWS::Lambda::WebFunctionEndpoint` recurso.

EntryPoint

El archivo que ejecuta el motor de ejecución, en relación con la raíz del paquete de despliegue. AWS SAM lo pasa al motor de ejecución de la variable de `AWS_LAMBDA_NODEJS_ENTRYPOINT` entorno reservada.

No puede usar esta propiedad cuando `EnvironmentVariables` es una función intrínseca.

Tipo: cadena

Obligatorio: no

Valor predeterminado: `index.js`

CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un CloudFormation equivalente.

EnvironmentVariables

Key-value pares para establecerlos como variables de entorno para la revisión.

- Las claves deben empezar por una letra y contener solo letras, números y guiones bajos.
- Los valores no deben estar vacíos y tener un máximo de 4.096 caracteres.
- AWS SAM convierte valores numéricos y booleanos en cadenas.

Tipo: mapa

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [ServiceConfig.EnvironmentVariables](#) propiedad de un `AWS::Lambda::WebFunctionRevision` recurso.

ExecutionRoleArn

El ARN de la función de IAM que asume la función cuando se ejecuta. Si no especificas un rol, AWS SAM crea uno con la política `AWSLambdaBasicExecutionRole` administrada y un identificador lógico de. *<resource-logical-id>Role*

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [ServiceConfig.ExecutionRoleArn](#) propiedad de un `AWS::Lambda::WebFunctionRevision` recurso.

FunctionName

El nombre de la función. Los nombres de las funciones deben ser únicos en una cuenta en todas las regiones. Si no especificas un nombre, AWS SAM genera uno a partir del identificador lógico y el identificador de pila, con un máximo de 64 caracteres. Al cambiar el nombre, se sustituyen la función, su revisión y su punto final.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [FunctionName](#) propiedad de los `AWS::Lambda::WebFunctionEndpoint` recursos `AWS::Lambda::WebFunction` y `AWS::Lambda::WebFunctionRevision`, y.

InlineCode

No se admite para `AWS::Serverless::WebFunction`. Utilice [CodeUri](#).

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

KmsKeyArn

El ARN de la AWS KMS clave utilizada para cifrar las variables de entorno de la revisión.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [KmsKeyArn](#) propiedad de un `AWS::Lambda::WebFunctionRevision` recurso.

LoggingConfig

La configuración de registro de la revisión, como el grupo de registros y los niveles de registro.

Tipo: mapa

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [ServiceConfig.TelemetryConfig.LoggingConfig](#) propiedad de un `AWS::Lambda::WebFunctionRevision` recurso.

MaxConcurrencyPerEnvironment

El número máximo de solicitudes simultáneas que gestiona cada entorno de ejecución.

Tipo: entero

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [ServiceConfig.MaxConcurrencyPerEnvironment](#) propiedad de un `AWS::Lambda::WebFunctionRevision` recurso.

ReplicaRegions

Las regiones en las que se va a replicar la función. Utilice esta propiedad con `MultiRegionPerRegion` puntos finales. Especifique una lista de códigos de región y grupos de regiones, o un objeto con `IncludeRegions` una opción `ExcludeRegions`.

Tipo: Lista | [ReplicaRegions](#)

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es similar a la [Regions](#) propiedad de un `AWS::Lambda::WebFunctionEndpoint` recurso. AWS SAM expande los grupos de regiones y se aplica `ExcludeRegions` para generar la lista de regiones.

RevisionDescription

Una descripción de la revisión que crea este despliegue.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [Description](#) propiedad de un `AWS::Lambda::WebFunctionRevision` recurso.

RevisionWeights

Un mapa de los identificadores de revisión y los pesos de tráfico. Utilice esta propiedad cuando `AutoDeploymentMode` esté `Disabled`. Especifique hasta dos revisiones. Cada peso es un número entero comprendido entre 1 y 100 y los pesos deben sumar 100.

Tipo: [RevisionWeights](#)

Obligatorio: condicional. Se requiere cuando `AutoDeploymentMode` es `Disabled` y `EndpointType` es `HomeRegion`.

CloudFormation compatibilidad: esta propiedad es similar a la [RevisionWeights](#) propiedad de un `AWS::Lambda::WebFunctionEndpoint` recurso. AWS SAM convierte el mapa en una lista de `Weight` objetos `RevisionId` y.

Runtime

El tiempo de ejecución de la función, por ejemplo `nodejs24.x`.

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [BuildConfig.RuntimeConfig.Runtime](#) propiedad de un `AWS::Lambda::WebFunctionRevision` recurso.

ScalingConfig

La configuración de escalado del punto final, como el número máximo de entornos de ejecución.

Tipo: mapa

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [ScalingConfig](#) propiedad de un `AWS::Lambda::WebFunctionEndpoint` recurso.

Tags

Un mapa de pares clave-valor que especifica las etiquetas agregadas a la función y al rol de ejecución que se AWS SAM genera.

Tipo: mapa

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es similar a la [Tags](#) propiedad de un `AWS::Lambda::WebFunction` recurso. La Tags propiedad en se AWS SAM compone de pares clave-valor (mientras que en CloudFormation esta propiedad se compone de una lista de Tag objetos). Además, agrega AWS SAM automáticamente una `lambda:createdBy:SAM` etiqueta a la función y al rol predeterminado que se genera para ella.

ThrottleConfig

La configuración de regulación del punto final, como la tasa máxima de solicitudes.

Tipo: mapa

Obligatorio: no

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [ThrottleConfig](#) propiedad de un `AWS::Lambda::WebFunctionEndpoint` recurso.

Timeout

El tiempo máximo, en segundos, que puede ejecutarse la función por solicitud. Los valores válidos oscilan entre 3 y 900.

Tipo: entero

Obligatorio: no

Valor predeterminado: 30

CloudFormation compatibilidad: esta propiedad se pasa directamente a la [ServiceConfig.TimeoutSeconds](#) propiedad de un `AWS::Lambda::WebFunctionRevision` recurso.

valores devueltos

Ref.

Cuando se proporciona el ID lógico de este recurso a la función Ref intrínseca, devuelve el ARN de la función web. Cuando necesite el nombre de la función, utilice el valor de `FunctionName` en su lugar.

Para obtener más información sobre el uso de la función Ref, consulta [Ref](#) en la Guía del usuario de AWS CloudFormation .

Fn: GetAtt

`Fn::GetAtt` devuelve un valor para un atributo especificado de este tipo. A continuación se indican los atributos disponibles y los valores devueltos de muestra.

Para obtener más información acerca de cómo usar `Fn::GetAtt`, consulta [Fn::GetAtt](#) en la Guía del usuario de AWS CloudFormation .

`FunctionArn`

El nombre del recurso de Amazon (ARN) de la función web.

State

El estado actual de la función web.

StateReason

El motivo del estado actual de la función web.

CreatedAt

La hora en que se creó la función web.

UpdatedAt

La hora en que se actualizó la función web por última vez.

Para hacer referencia a la revisión o al punto final que se AWS SAM genera, utilice sus identificadores lógicos `<resource-logical-id>Revision` y `<resource-logical-id>Endpoint`. Por ejemplo, `!GetAtt MyWebFunctionEndpoint.DomainName` devuelve el nombre de dominio del punto final de una función web con el ID lógico `MyWebFunction`. Para obtener más información, consulte [CloudFormation recursos generados cuando AWS::Serverless::WebFunction se especifica](#).

Ejemplos

Función web básica

El siguiente ejemplo crea una función web con un IAM-authorized punto final en la región de la pila. AWS SAM genera la función de ejecución y el punto final enruta todo el tráfico a la revisión que crea esta implementación.

```
AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31

Resources:
  MyWebFunction:
    Type: AWS::Serverless::WebFunction
    Properties:
      Runtime: nodejs24.x
      AuthType: IamAuth
      CodeUri:
        Bucket: amzn-s3-demo-bucket
        Key: webfn.zip
        Version: object-version-id
```

```
Timeout: 30
```

```
Outputs:
```

```
  EndpointUrl:
```

```
    Value: !Sub https://${MyWebFunctionEndpoint.DomainName}
```

Divida el tráfico entre dos revisiones

El siguiente ejemplo deshabilita la implementación automática y envía el 90% del tráfico a una revisión existente y el 10% a otra. Ambas revisiones deben existir ya.

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

```
Transform: AWS::Serverless-2016-10-31
```

```
Resources:
```

```
  MyWebFunction:
```

```
    Type: AWS::Serverless::WebFunction
```

```
    Properties:
```

```
      Runtime: nodejs24.x
```

```
      AuthType: IamAuth
```

```
      CodeUri: s3://amzn-s3-demo-bucket/webfn-v2.zip
```

```
      AutoDeploymentMode: Disabled
```

```
      RevisionWeights:
```

```
        rev-abc123: 90
```

```
        rev-def456: 10
```

Multi-Region Función web

En el siguiente ejemplo, se replica la función en las regiones de EE. UU., excepto en las regiones de EE. UU. `us-west-1` y se enruta cada solicitud a la región más cercana. Como `RevisionWeights` se omite, el punto final enruta todo el tráfico a la revisión creada en esta pila.

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

```
Transform: AWS::Serverless-2016-10-31
```

```
Resources:
```

```
  MyWebFunction:
```

```
    Type: AWS::Serverless::WebFunction
```

```
    Properties:
```

```
      Runtime: nodejs24.x
```

```
      AuthType: IamAuth
```

```
      CodeUri:
```

```
        Bucket: amzn-s3-demo-bucket
```



```

    Key: webfn.zip
    Version: object-version-id
    EndpointName: prodmr
    EndpointType: MultiRegion
    AutoDeploymentMode: Disabled
    ReplicaRegions:
      IncludeRegions:
        - US
      ExcludeRegions:
        - us-west-1

```

Función web con salida de VPC

El siguiente ejemplo crea un conector de red para Web Functions y enruta el tráfico saliente de la función a través de las subredes y los grupos de seguridad del conector.

```

AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31

Resources:
  MyConnector:
    Type: AWS::Serverless::NetworkConnector
    Properties:
      ComputeTypes:
        - WebFunction
      VpcConfig:
        SubnetIds:
          - subnet-0abc123def4567890
          - subnet-0def456abc7890123
        SecurityGroupIds:
          - sg-0abc1234abcde0001
        NetworkProtocol: IPv4

  MyWebFunction:
    Type: AWS::Serverless::WebFunction
    Properties:
      Runtime: nodejs24.x
      AuthType: IamAuth
      CodeUri:
        Bucket: amzn-s3-demo-bucket
        Key: webfn.zip
        Version: object-version-id
      EgressNetworkConnectorArn: !GetAtt MyConnector.LatestVersionArn

```

CodeUri

Especifica la ubicación en Amazon S3 del paquete de implementación de una función web. Puede especificar la ubicación como un objeto o como una cadena del formulario `s3://bucket/key` con un parámetro de `?versionId=version` consulta opcional.

El bucket debe tener habilitado el control de versiones y una política de bucket que conceda `lambda.amazonaws.com` los `s3:GetObjectVersion` permisos `s3:GetObject` y. Le recomendamos que defina la política con una `aws:SourceArn` condición que coincida con las revisiones de la función. Si el objeto está cifrado con SSE-KMS, utilice una clave gestionada por el cliente cuya política de claves permita `lambda.amazonaws.com` realizar `llamaskms:Decrypt`. No se admite la clave AWS administrada de Amazon S3 (`aws/s3`). Para obtener más información, consulte [Implementación del código de funciones web de Amazon S3](#) en la Guía para AWS Lambda desarrolladores.

Sintaxis

Para declarar esta entidad en su plantilla AWS Serverless Application Model (AWS SAM), utilice la siguiente sintaxis.

YAML

```
Bucket: String  
Key: String  
Version: String
```

Propiedades

Bucket

El nombre del bucket de Amazon S3 que contiene el paquete de implementación.

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se transfiere directamente a la Bucket propiedad [BuildConfig.CodeConfig.S3Object](#) de un `AWS::Lambda::WebFunctionRevision` recurso.

Key

La clave de Amazon S3 del paquete de implementación.

Tipo: cadena

Obligatorio: sí

CloudFormation compatibilidad: esta propiedad se pasa directamente a la Key propiedad [BuildConfig.CodeConfig.S3Object](#) de un `AWS::Lambda::WebFunctionRevision` recurso.

Version

El identificador de versión del objeto del paquete de despliegue. Especifique una versión para fijar la revisión a una carga específica.

Tipo: cadena

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es similar a la `VersionId` propiedad [BuildConfig.CodeConfig.S3Object](#) de un `AWS::Lambda::WebFunctionRevision` recurso. AWS SAM pasa el valor como `VersionId`.

Ejemplos

Forma de objeto

```
CodeUri:
  Bucket: amzn-s3-demo-bucket
  Key: webfn.zip
  Version: object-version-id
```

Formulario URI

```
CodeUri: s3://amzn-s3-demo-bucket/webfn.zip?versionId=object-version-id
```

ReplicaRegions

Especifica las regiones en las que un punto final `MultiRegion` o una función `PerRegion` web replica la función. Puede especificar una lista de códigos de región y grupos de regiones o un objeto que incluya regiones y, de forma opcional, excluya algunas de ellas.

Puede usar los siguientes grupos de regiones en cualquier lugar donde especifique una región:

- US: us-east-1, us-east-2, us-west-1, us-west-2
- EU: eu-west-1, eu-west-2, eu-west-3, eu-central-1, eu-north-1
- AP: ap-south-1, ap-northeast-1, ap-northeast-2, ap-northeast-3, ap-southeast-1, ap-southeast-2
- CA: ca-central-1
- SA: sa-east-1
- SPARSE_GLOBAL: una o dos regiones por continente para una cobertura global con menos regiones: us-east-1, eu-west-1, eu-central-1, ap-northeast-1, ap-southeast-1, ap-southeast-2, sa-east-1
- ALL_STANDARD: todas las regiones de los SA grupos US EUAP, CA,, y (17 regiones)

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
ExcludeRegions: List  
IncludeRegions: List
```

Propiedades

ExcludeRegions

Una lista de códigos de región y grupos de regiones para eliminarlos `IncludeRegions`. También debe especificarlo `IncludeRegions`. Las entradas no pueden ser funciones intrínsecas y no se pueden usar `ExcludeRegions` cuando `IncludeRegions` contienen funciones intrínsecas.

Tipo: lista

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un CloudFormation equivalente.

IncludeRegions

Una lista de códigos de región y grupos de regiones en los que replicar la función. La lista en sí misma no puede ser una función intrínseca, pero las entradas individuales sí.

Tipo: lista

Obligatorio: no

CloudFormation compatibilidad: esta propiedad es similar a la [Regions](#) propiedad de un `AWS::Lambda::WebFunctionEndpoint` recurso. AWS SAM expande los grupos de regiones y elimina cualquier `ExcludeRegions` entrada para generar la lista de regiones.

Ejemplos

Lista de regiones

En el siguiente ejemplo, se replica la función en dos regiones.

```
ReplicaRegions:
  - us-east-1
  - eu-west-1
```

Grupo de regiones con exclusiones

El siguiente ejemplo replica la función en las regiones de EE. UU., excepto. `us-west-1`

```
ReplicaRegions:
  IncludeRegions:
    - US
  ExcludeRegions:
    - us-west-1
```

RevisionWeights

Especifica cómo el punto final de una función web divide el tráfico entre las revisiones. Utilice esta propiedad cuando `AutoDeploymentMode` esté `Disabled`, por ejemplo, para ejecutar un Canary o una blue/green implementación.

`RevisionWeights` es un mapa. Cada clave es un identificador de revisión y cada valor es el porcentaje de tráfico que se dirige a esa revisión. Se aplican las siguientes reglas:

- Especifique como máximo dos revisiones.
- Cada peso es un número entero comprendido entre 1 y 100 y los pesos deben sumar 100.
- Los pesos individuales pueden ser funciones intrínsecas, pero el mapa en sí no puede ser una función intrínseca.

Tipo: mapa

CloudFormation compatibilidad: esta propiedad es similar a la [RevisionWeights](#) propiedad de un `AWS::Lambda::WebFunctionEndpoint` recurso. AWS SAM convierte el mapa en una lista de objetos, cada uno con una `RevisionId` y una `Weight`.

Sintaxis

Para declarar esta entidad en tu plantilla AWS Serverless Application Model (AWS SAM), usa la siguiente sintaxis.

YAML

```
revision-id: Integer
revision-id: Integer
```

Ejemplos

División en Canarias

El siguiente ejemplo envía el 90% del tráfico a `rev-abc123` y el 10% a `rev-def456`.

```
AutoDeploymentMode: Disabled
RevisionWeights:
  rev-abc123: 90
  rev-def456: 10
```

AWS SAM lo transforma en la siguiente propiedad de punto final:

```
RevisionWeights:
  - RevisionId: rev-abc123
    Weight: 90
  - RevisionId: rev-def456
    Weight: 10
```

Generado CloudFormation recursos para AWS SAM

En esta sección se proporcionan detalles sobre los CloudFormation recursos que se crean al AWS SAM procesar la AWS plantilla. El conjunto de CloudFormation recursos que se AWS SAM genera varía en función de los escenarios que especifique. Un escenario es la combinación de recursos y propiedades de AWS SAM especificada en el archivo de plantilla. Puede hacer referencia a los recursos de CloudFormation generados en otros lugares del archivo de plantilla, de forma similar a como hace referencia a los recursos que declara explícitamente en el archivo de plantilla.

Por ejemplo, si especifica un recurso de `AWS::Serverless::Function` en el archivo de plantillas de AWS SAM, AWS SAM siempre generará un recurso base `AWS::Lambda::Function`. Si también especifica la `AutoPublishAlias` propiedad opcional, generará `AWS::Lambda::Alias` y `AWS::Lambda::Version` recursos AWS SAM adicionales.

En esta sección se enumeran los escenarios y los CloudFormation recursos que generan, y se muestra cómo hacer referencia a CloudFormation los recursos generados en el archivo de AWS SAM plantilla.

Referencia generada CloudFormation recursos

Tiene dos opciones para hacer referencia a CloudFormation los recursos generados en su archivo de AWS SAM plantilla: por propiedad referenciable `LogicalId` o por propiedad referenciable.

Referenciación generada CloudFormation recursos por `LogicalId`

Los CloudFormation recursos que AWS SAM genera cada uno tienen un [LogicalId](#), que es un identificador alfanumérico (A-Z, a-z, 0-9) que es único dentro de un archivo de plantilla. AWS SAM utiliza los AWS SAM recursos `LogicalIds` del archivo de plantilla para construir `LogicalIds` los CloudFormation recursos que genera. Puede usar el `LogicalId` CloudFormation recurso generado para acceder a las propiedades de ese recurso dentro de su archivo de plantilla, del mismo modo que lo haría con un CloudFormation recurso que haya declarado explícitamente. Para obtener más información sobre `LogicalIds` las AWS SAM plantillas CloudFormation y las plantillas, consulte [los recursos](#) en la Guía del AWS CloudFormation usuario.

Note

Algunos `LogicalIds` de los recursos generados incluyen un valor hash único para evitar conflictos con los espacios de los nombres. Los `LogicalIds` de estos recursos se obtienen

cuando se crea la pila. Puede recuperarlos solo después de haber creado la pila con el Consola de administración de AWS, AWS CLI, o uno de los AWS SDK. No recomendamos referenciar estos recursos con `LogicalId` porque los valores hash pueden cambiar.

Referenciación generada CloudFormation recursos por propiedad referenciable

Para algunos recursos generados, AWS SAM proporciona una propiedad referenciable del recurso. AWS SAM Puede utilizar esta propiedad para hacer referencia a un CloudFormation recurso generado y a sus propiedades en el archivo de AWS SAM plantilla.

Note

No todos los CloudFormation recursos generados tienen propiedades a las que se puede hacer referencia. Para dichos recursos, debe utilizar los `LogicalId`.

Generado CloudFormation escenarios de recursos

En la siguiente tabla se resumen los AWS SAM recursos y las propiedades que componen los escenarios que generan CloudFormation recursos. Los temas de la columna Escenarios proporcionan detalles sobre los CloudFormation recursos adicionales que se AWS SAM generan para ese escenario.

AWS SAM recurso	CloudFormation Recurso base	Escenarios
<code>AWS::Serverless::Api</code>	<code>AWS::ApiGateway::RestApi</code>	<ul style="list-style-type: none"> DomainName se especifica la propiedad UsagePlan se especifica la propiedad
<code>AWS::Serverless::Application</code>	<code>AWS::CloudFormation::Stack</code>	<ul style="list-style-type: none"> Además de generar el CloudFormation recurso base, no hay escenarios adicionales para este recurso sin servidor.

AWS SAM recurso	CloudForm ation Recurso base	Escenarios
<u>AWS::Serv erless::C apacityPr ovider</u>	<u>AWS::Lamb da::Capac ityProvid er</u>	<ul style="list-style-type: none"> • <u>OperatorRole la propiedad no está especificada</u>
<u>AWS::Serv erless::M icrovmIma ge</u>	<u>AWS::Lamb da::Micro vmImage</u>	<ul style="list-style-type: none"> • <u>BuildRoleArn la propiedad no está especificada</u>
<u>AWS::Serv erless::N etworkCon nector</u>	<u>AWS::Lamb da::Netwo rkConnect or</u>	<ul style="list-style-type: none"> • <u>OperatorRole la propiedad no está especificada</u>
<u>AWS::Serv erless::W ebFuncio n</u>	<u>AWS::Lamb da::WebFu nction</u>	<ul style="list-style-type: none"> • <u>ExecutionRoleArn la propiedad no está especificada</u>

AWS SAM recurso	CloudFormation Recurso base	Escenarios
<u>AWS::Serverless::Function</u>	<u>AWS::Lambda::Function</u>	<ul style="list-style-type: none"> • <u>AutoPublishAlias se especifica la propiedad</u> • <u>La propiedad del rol no está especificada</u> • <u>DeploymentPreference se especifica la propiedad</u> • <u>Se ha especificado una fuente de eventos de API</u> • <u>Se ha especificado una fuente de HttpApi eventos</u> • <u>Se ha especificado una origen de eventos de transmisión</u> • <u>Se ha especificado una origen de eventos de puente de eventos (o bus de eventos)</u> • <u>Se ha especificado una fuente de IoTRule eventos</u> • <u>OnSuccess (o OnFailure) la propiedad está especificada para los eventos de Amazon SNS</u> • <u>OnSuccess (o OnFailure) la propiedad está especificada para los eventos de Amazon SQS</u>
<u>AWS::Serverless::HttpApi</u>	<u>AWS::ApiGatewayV2::Api</u>	<ul style="list-style-type: none"> • <u>StageName se especifica la propiedad</u> • <u>StageName la propiedad no está especificada</u> • <u>DomainName se especifica la propiedad</u>
<u>AWS::Serverless::LayerVersion</u>	<u>AWS::Lambda::LayerVersion</u>	<ul style="list-style-type: none"> • Además de generar el CloudFormation recurso base, no hay escenarios adicionales para este recurso sin servidor.

AWS SAM recurso	CloudFormation Recurso base	Escenarios
<u>AWS::Serverless::SimpleTable</u>	<u>AWS::DynamoDB::Table</u>	<ul style="list-style-type: none"> Además de generar el CloudFormation recurso base, no hay escenarios adicionales para este recurso sin servidor.
<u>AWS::Serverless::StateMachine</u>	<u>AWS::StepFunctions::StateMachine</u>	<ul style="list-style-type: none"> <u>La propiedad del rol no está especificada</u> <u>Se ha especificado un origen de eventos de API</u> <u>Se ha especificado una origen de eventos de puente de eventos (o bus de eventos)</u>

Temas

- [CloudFormation recursos generados cuando AWS::Serverless::Api se especifica](#)
- [CloudFormation recursos generados cuando AWS::Serverless::Application se especifica](#)
- [CloudFormation recursos generados cuando AWS::Serverless::CapacityProvider se especifica](#)
- [CloudFormation recursos generados cuando se especifica AWS::Serverless::Connector](#)
- [CloudFormation recursos generados cuando AWS::Serverless::Function se especifica](#)
- [CloudFormation recursos generados cuando AWS::Serverless::GraphQLApi se especifica](#)
- [CloudFormation recursos generados cuando AWS::Serverless::HttpApi se especifica](#)
- [CloudFormation recursos generados cuando AWS::Serverless::MicrovmImage se especifica](#)
- [CloudFormation recursos generados cuando AWS::Serverless::NetworkConnector se especifica](#)
- [CloudFormation recursos generados cuando AWS::Serverless::WebFunction se especifica](#)
- [CloudFormation recursos generados cuando AWS::Serverless::WebSocketApi se especifica](#)
- [CloudFormation recursos generados cuando AWS::Serverless::LayerVersion se especifica](#)
- [CloudFormation recursos generados cuando AWS::Serverless::SimpleTable se especifica](#)
- [CloudFormation recursos generados cuando AWS::Serverless::StateMachine se especifica](#)

CloudFormation recursos generados cuando AWS::Serverless::Api se especifica

Cuando AWS::Serverless::Api se especifica un, AWS Serverless Application Model (AWS SAM) siempre genera un CloudFormation recurso AWS::ApiGateway::RestApi base. Además, siempre genera un recurso AWS::ApiGateway::Stage y un recurso AWS::ApiGateway::Deployment.

AWS::ApiGateway::RestApi

LogicalId: *<api-LogicalId>*

Propiedad referenciable: N/A (debe usar la LogicalId para hacer referencia a este CloudFormation recurso)

AWS::ApiGateway::Stage

LogicalId: *<api-LogicalId><stage-name>Stage*

<stage-name> es la cadena en la que está establecida la propiedad StageName. Por ejemplo, si establece StageName como Gamma, entonces LogicalId es *MyRestApiGammaStage*.

Propiedad referenciable: *<api-LogicalId>.Stage*

AWS::ApiGateway::Deployment

LogicalId: *<api-LogicalId>Deployment<sha>*

<sha> es un valor hash único que se genera cuando se crea la pila. Por ejemplo, *MyRestApiDeployment926eeb5ff1*.

Propiedad referenciable: *<api-LogicalId>.Deployment*

Además de estos CloudFormation recursos, cuando AWS::Serverless::Api se especifica, AWS SAM genera CloudFormation recursos adicionales para los siguientes escenarios.

Escenarios

- [DomainName se especifica la propiedad](#)
- [UsagePlan se especifica la propiedad](#)

DomainName se especifica la propiedad

Cuando se especifica la DomainName Domain propiedad de AWS::Serverless::Api una, AWS SAM genera el AWS::ApiGateway::DomainName CloudFormation recurso.

AWS::ApiGateway::DomainName

LogicalId: ApiGatewayDomainName<sha>

<sha> es un valor hash único que se genera cuando se crea la pila. Por ejemplo:
ApiGatewayDomainName926eeb5ff1.

Propiedad referenciable: <api-LogicalId>.DomainName

UsagePlan se especifica la propiedad

Cuando se especifica la Auth propiedad de AWS::Serverless::Api una, AWS SAM genera los siguientes CloudFormation recursos:
AWS::ApiGateway::UsagePlanAWS::ApiGateway::UsagePlanKey,
yAWS::ApiGateway::ApiKey. UsagePlan

AWS::ApiGateway::UsagePlan

LogicalId: <api-LogicalId>UsagePlan

Propiedad referenciable: <api-LogicalId>.UsagePlan

AWS::ApiGateway::UsagePlanKey

LogicalId: <api-LogicalId>UsagePlanKey

Propiedad referenciable: <api-LogicalId>.UsagePlanKey

AWS::ApiGateway::ApiKey

LogicalId: <api-LogicalId>ApiKey

Propiedad referenciable: <api-LogicalId>.ApiKey

CloudFormation recursos generados cuando `AWS::Serverless::Application` se especifica

Cuando `AWS::Serverless::Application` se especifica un, AWS Serverless Application Model (AWS SAM) genera un CloudFormation recurso `AWS::CloudFormation::Stack` base.

`AWS::CloudFormation::Stack`

LogicalId: `<application-LogicalId>`

Propiedad referenciable: N/A (debe usar la `LogicalId` para hacer referencia a este CloudFormation recurso)

CloudFormation recursos generados cuando `AWS::Serverless::CapacityProvider` se especifica

Cuando se especifica `AWS::Serverless::CapacityProvider`, AWS Serverless Application Model (AWS SAM) genera un recurso `AWS::Lambda::CapacityProvider` de base CloudFormation .

`AWS::Lambda::CapacityProvider`

LogicalId: `<capacityprovider-LogicalId>`

Propiedad referenciable: N/A (debe usar la `LogicalId` para hacer referencia a este CloudFormation recurso)

Además de este CloudFormation recurso, cuando `AWS::Serverless::CapacityProvider` se especifica, AWS SAM también genera CloudFormation recursos para los siguientes escenarios:

Escenarios

- [OperatorRole la propiedad no está especificada](#)

OperatorRole la propiedad no está especificada

Cuando no `AWS::Serverless::CapacityProvider` se especifica la `OperatorRole` propiedad de un, AWS SAM genera un `AWS::IAM::Role` CloudFormation recurso con la política `AWSLambdaManagedEC2ResourceOperator` administrada adjunta.

AWS::IAM::Role

LogicalId: <capacityprovider-LogicalId>OperatorRole

Propiedad referenciable: N/A (debe usar la LogicalId para hacer referencia a este CloudFormation recurso)

CloudFormation recursos generados cuando se especifica AWS::Serverless::Connector

Note

Al definir los conectores a través de la propiedad `Connectors` incrustada, primero se transforma en un recurso `AWS::Serverless::Connector` antes de generar estos recursos.

Al especificar un `AWS::Serverless::Connector` recurso en una AWS SAM plantilla, AWS SAM genera los siguientes AWS CloudFormation recursos según sea necesario.

AWS::IAM::ManagedPolicy

LogicalId: <connector-LogicalId>Policy

Propiedad referenciable: N/A (Para hacer referencia a este CloudFormation recurso, debe usar la LogicalId.)

AWS::SNS::TopicPolicy

LogicalId: <connector-LogicalId>TopicPolicy

Propiedad referenciable: N/A (Para hacer referencia a este CloudFormation recurso, debe usar la LogicalId)

AWS::SQS::QueuePolicy

LogicalId: <connector-LogicalId>QueuePolicy

Propiedad referenciable: N/A (Para hacer referencia a este CloudFormation recurso, debe usar la LogicalId)

AWS::Lambda::Permission

LogicalId: **<connector-LogicalId><permission>** LambdaPermission

<permission> es un permiso especificado por la propiedad Permissions. Por ejemplo, Write.

Propiedad referenciable: N/A (Para hacer referencia a este CloudFormation recurso, debe usar la.) LogicalId

CloudFormation recursos generados cuando AWS::Serverless::Function se especifica

Cuando AWS::Serverless::Function se especifica un, AWS Serverless Application Model (AWS SAM) siempre crea un CloudFormation recurso AWS::Lambda::Function base.

AWS::Lambda::Function

LogicalId: **<function-LogicalId>**

Propiedad referenciable: N/A (debe usar la LogicalId para hacer referencia a este CloudFormation recurso)

Además de este CloudFormation recurso, cuando AWS::Serverless::Function se especifica, AWS SAM también genera CloudFormation recursos para los siguientes escenarios.

Escenarios

- [Propiedades de funciones principales](#)
- [Orígenes de eventos](#)
- [Configuración de eventos](#)

Propiedades de funciones principales

Los siguientes escenarios generan CloudFormation recursos en función de las propiedades de las funciones principales:

La propiedad del rol no está especificada

Cuando no `AWS::Serverless::Function` se especifica la `Role` propiedad de un, AWS SAM genera un `AWS::IAM::Role` CloudFormation recurso.

AWS::IAM::Role

LogicalId: `<function-LogicalId>Role`

Propiedad referenciable: N/A (debe usar la `LogicalId` para hacer referencia a este CloudFormation recurso)

`AutoPublishAlias` se especifica la propiedad

Cuando `AWS::Serverless::Function` se especifica la `AutoPublishAlias` propiedad de un, AWS SAM genera los siguientes CloudFormation recursos: `AWS::Lambda::Alias` y `AWS::Lambda::Version`.

AWS::Lambda::Alias

LogicalId: `<function-LogicalId>Alias<alias-name>`

`<alias-name>` es la cadena que se establece en `AutoPublishAlias`. Por ejemplo, si se establece `AutoPublishAlias` en `live`, el `LogicalId` es: `MyFunction Aliaslive`.

Propiedad referenciable: `<function-LogicalId>.Alias`

AWS::Lambda::Version

LogicalId: `<function-LogicalId>Version<sha>`

`<sha>` es un valor hash único que se genera cuando se crea la pila. Por ejemplo, `MyFunction Version926eeb5ff1`.

Propiedad referenciable: `<function-LogicalId>.Version`

Para obtener información adicional sobre la `AutoPublishAlias` propiedad, consulte la [sección Propiedades de AWS::Serverless::Function](#).

`DeploymentPreference` se especifica la propiedad

Cuando `AWS::Serverless::Function` se especifica la `DeploymentPreference` propiedad de un, AWS SAM genera los siguientes CloudFormation recursos: `AWS::CodeDeploy::Application`

y `AWS::CodeDeploy::DeploymentGroup`. Además, si no se especifica la `Role` propiedad del `DeploymentPreference` objeto, AWS SAM también genera un `AWS::IAM::Role` CloudFormation recurso.

AWS::CodeDeploy::Application

LogicalId: `ServerlessDeploymentApplication`

Propiedad referenciable: N/A (debe usar la `LogicalId` para hacer referencia a este CloudFormation recurso)

AWS::CodeDeploy::DeploymentGroup

LogicalId: `<function-LogicalId>DeploymentGroup`

Propiedad referenciable: N/A (debe usar la para hacer referencia `LogicalId` a este recurso) CloudFormation

AWS::IAM::Role

LogicalId: `CodeDeployServiceRole`

Propiedad referenciable: N/A (debe usar la para hacer referencia `LogicalId` a este recurso) CloudFormation

`FunctionUrlConfig` se especifica la propiedad

Cuando se especifica la `FunctionUrlConfig` propiedad, AWS SAM genera diferentes CloudFormation recursos en función de `AuthType`.

Cuando `AuthType`: `NONE` se especifica, AWS SAM genera los siguientes CloudFormation recursos:

AWS::Lambda::Permission (Invocar acceso)

LogicalId: `<function-LogicalId>URLInvokeAllowPublicAccess`

Propiedad referenciable: N/A (debe usar la `LogicalId` para hacer referencia a este CloudFormation recurso)

AWS::Lambda::Permission (Acceso público)

LogicalId: `<function-LogicalId>UrlPublicPermissions`

Propiedad referenciable: N/A (debe usar la para hacer referencia LogicalId a este recurso)
CloudFormation

AWS::Lambda::Url

LogicalId: *<function-LogicalId>Url*

Propiedad referenciable: N/A (debe usar la para hacer referencia LogicalId a este recurso)
CloudFormation

Cuando AuthType: AWS_IAM se especifica, solo AWS SAM genera:

AWS::Lambda::Url

LogicalId: *<function-LogicalId>Url*

Propiedad referenciable: N/A (debe usar la LogicalId para hacer referencia a este
CloudFormation recurso)

Para obtener información adicional sobre la FunctionUrlConfig propiedad, consulte.

[FunctionUrlConfig](#)

Orígenes de eventos

Los siguientes escenarios generan CloudFormation recursos en función de las fuentes de eventos:

Se ha especificado una fuente de eventos de API

Cuando la Event propiedad de an AWS::Serverless::Function se establece enApi, pero no se especifica, se AWS SAM genera el AWS::ApiGateway::RestApi CloudFormation recurso.
RestApiId

AWS::ApiGateway::RestApi

LogicalId: ServerlessRestApi

Propiedad referenciable: N/A (debe usar la LogicalId para hacer referencia a este
CloudFormation recurso)

Se ha especificado una fuente de HttpApi eventos

Cuando la Event propiedad de un `AWS::Serverless::Function` se establece en `HttpApi`, pero no se especifica, AWS SAM genera el `AWS::ApiGatewayV2::Api` CloudFormation recurso. `ApiId`

AWS::ApiGatewayV2::Api

LogicalId: `ServerlessHttpApi`

Propiedad referenciable: N/A (debe usar la `LogicalId` para hacer referencia a este CloudFormation recurso)

Se ha especificado una origen de eventos de transmisión

Cuando la Event propiedad de un `AWS::Serverless::Function` se establece en uno de los tipos de transmisión, AWS SAM genera el `AWS::Lambda::EventSourceMapping` CloudFormation recurso. Esto se aplica a los siguientes tipos: `DynamoDB`, `Kinesis`, `MQ`, `MSK` y `SQS`.

AWS::Lambda::EventSourceMapping

LogicalId: `<function-LogicalId><event-LogicalId>`

Propiedad referenciable: N/A (debe usar la `LogicalId` para hacer referencia a este CloudFormation recurso)

Se ha especificado una origen de eventos de puente de eventos (o bus de eventos)

Cuando la Event propiedad de un `AWS::Serverless::Function` se establece en uno de los tipos de puente de eventos (o bus de eventos), AWS SAM genera el `AWS::Events::Rule` CloudFormation recurso. Esto se aplica a los siguientes tipos: `EventBridgeRule`, `Schedule`, y `CloudWatchEvents`.

AWS::Events::Rule

LogicalId: `<function-LogicalId><event-LogicalId>`

Propiedad referenciable: N/A (debe usar la `LogicalId` para hacer referencia a este CloudFormation recurso)

Se ha especificado una fuente de IoTRule eventos

Cuando la Event propiedad de un `AWS::Serverless::Function` se establece en `IoTRule`, AWS SAM genera el `AWS::IoT::TopicRule` CloudFormation recurso.

AWS::IoT::TopicRule

LogicalId: `<function-LogicalId><event-LogicalId>`

Propiedad referenciable: N/A (debe usar la `LogicalId` para hacer referencia a este CloudFormation recurso)

Configuración de eventos

Los siguientes escenarios generan CloudFormation recursos en función de la configuración del evento:

OnSuccess (o OnFailure) la propiedad está especificada para los eventos de Amazon SNS

Cuando se especifica la OnSuccess (o OnFailure) DestinationConfig propiedad de la EventInvokeConfig propiedad de un `AWS::Serverless::Function` y el tipo de destino es SNS pero no se especifica el ARN de destino, AWS SAM genera los siguientes CloudFormation recursos: `AWS::Lambda::EventInvokeConfig` y `AWS::SNS::Topic`

AWS::Lambda::EventInvokeConfig

LogicalId: `<function-LogicalId>EventInvokeConfig`

Propiedad referenciable: N/A (debe usar la `LogicalId` para hacer referencia a este recurso) CloudFormation

AWS::SNS::Topic

LogicalId: `<function-LogicalId>OnSuccessTopic` (o `<function-LogicalId>OnFailureTopic`)

Propiedad referenciable: `<function-LogicalId>.DestinationTopic`

Si ambos OnSuccess y OnFailure se especifican para un evento de Amazon SNS, para distinguir entre los recursos generados, debe usar el `LogicalId`.

OnSuccess (o OnFailure) la propiedad está especificada para los eventos de Amazon SQS

Cuando se especifica la OnSuccess (o OnFailure) DestinationConfig propiedad de la EventInvokeConfig propiedad de un AWS::Serverless::Function y el tipo de destino es SQS pero no se especifica el ARN de destino, AWS SAM genera los siguientes CloudFormation recursos: AWS::Lambda::EventInvokeConfig y AWS::SQS::Queue

AWS::Lambda::EventInvokeConfig

LogicalId: <function-LogicalId>EventInvokeConfig

Propiedad referenciable: N/A (debe usar la LogicalId para hacer referencia a este recurso)
CloudFormation

AWS::SQS::Queue

*LogicalId: <function-LogicalId>OnSuccessQueue (o
<function-LogicalId>OnFailureQueue)*

Propiedad referenciable: *<function-LogicalId>.DestinationQueue*

Si ambos OnSuccess y OnFailure se especifican para un evento de Amazon SQS, para distinguir entre los recursos generados, debe utilizar el LogicalId.

CloudFormation recursos generados cuando AWS::Serverless::GraphQLApi se especifica

Cuando especificas un AWS::Serverless::GraphQLApi recurso en una plantilla AWS Serverless Application Model (AWS SAM), AWS SAM siempre crea los siguientes AWS CloudFormation recursos base.

AWS::AppSync::DataSource

LogicalId: <graphqlapi-LogicalId><datasource-RelativeId><datasource-Type>DataSource

Propiedad referenciable: N/A (debe usar la LogicalId para hacer referencia a este CloudFormation recurso)

AWS::AppSync::FunctionConfiguration

LogicalId: <graphqlapi-LogicalId><function-RelativeId>

Propiedad referenciable: N/A (debe usar la para hacer referencia LogicalId a este recurso)
CloudFormation

AWS::AppSync::GraphQLApi

LogicalId: <graphqlapi-LogicalId>

Propiedad referenciable: N/A (debe usar la para hacer referencia LogicalId a este recurso)
CloudFormation

AWS::AppSync::GraphQLSchema

LogicalId: <graphqlapi-LogicalId>Schema

Propiedad referenciable: N/A (debe usar la para hacer referencia LogicalId a este recurso)
CloudFormation

AWS::AppSync::Resolver

LogicalId: <graphqlapi-LogicalId><OperationType><resolver-RelativeId>

Propiedad referenciable: N/A (debe usar la para hacer referencia LogicalId a este recurso)
CloudFormation

Además de estos CloudFormation recursos, cuando se especifique, también

AWS::Serverless::GraphQLApi se AWS SAM pueden generar los siguientes CloudFormation recursos.

AWS::AppSync::ApiCache

LogicalId: <graphqlapi-LogicalId>ApiCache

Propiedad referenciable: N/A (debe utilizar la LogicalId para hacer referencia a este
CloudFormation recurso)

AWS::AppSync::ApiKey

LogicalId: <graphqlapi-LogicalId><apikey-RelativeId>

Propiedad referenciable: N/A (debe usar la para hacer referencia LogicalId a este recurso)
CloudFormation

AWS::AppSync::DomainName

LogicalId: <graphqlapi-LogicalId>DomainName

Propiedad referenciable: N/A (debe usar la para hacer referencia LogicalId a este recurso)
CloudFormation

AWS::AppSync::DomainNameApiAssociation

LogicalId: <graphqlapi-LogicalId>DomainNameApiAssociation

Propiedad referenciable: N/A (debe usar la para hacer referencia LogicalId a este recurso)
CloudFormation

AWS SAM también puede usar el **AWS::Serverless::Connector** recurso para conceder permisos. Para obtener más información, consulte [CloudFormation recursos generados cuando se especifica AWS::Serverless::Connector](#).

CloudFormation recursos generados cuando AWS::Serverless::HttpApi se especifica

Cuando **AWS::Serverless::HttpApi** se especifica un, AWS Serverless Application Model (AWS SAM) genera un CloudFormation recurso **AWS::ApiGatewayV2::Api** base.

AWS::ApiGatewayV2::Api

LogicalId: <httpapi-LogicalId>

Propiedad referenciable: N/A (debe usar la LogicalId para hacer referencia a este CloudFormation recurso)

Además de este CloudFormation recurso, cuando **AWS::Serverless::HttpApi** se especifica, AWS SAM también genera CloudFormation recursos para los siguientes escenarios:

Escenarios

- [StageName se especifica la propiedad](#)
- [StageName la propiedad no está especificada](#)
- [DomainName se especifica la propiedad](#)

StageName se especifica la propiedad

Cuando **AWS::Serverless::HttpApi** se especifica la StageName propiedad de un, AWS SAM genera el **AWS::ApiGatewayV2::Stage** CloudFormation recurso.

AWS::ApiGatewayV2::Stage

LogicalId: *<httpapi-LogicalId><stage-name>*Stage

<stage-name> es la cadena en la que está establecida la propiedad StageName. Por ejemplo, si se establece StageName enGamma, el LogicalId es: *MyHttpApiGamma* Stage.

Propiedad referenciable: *<httpapi-LogicalId>*.Stage

StageName la propiedad no está especificada

Cuando no *AWS::Serverless::HttpApi* se especifica la StageName propiedad de un, AWS SAM genera el *AWS::ApiGatewayV2::Stage* CloudFormation recurso.

AWS::ApiGatewayV2::Stage

LogicalId: *<httpapi-LogicalId>*ApiGatewayDefaultStage

Propiedad referenciable: *<httpapi-LogicalId>*.Stage

DomainName se especifica la propiedad

Cuando se especifica la DomainName Domain propiedad de *AWS::Serverless::HttpApi* una, AWS SAM genera el *AWS::ApiGatewayV2::DomainName* CloudFormation recurso.

AWS::ApiGatewayV2::DomainName

LogicalId: ApiGatewayDomainNameV2*<sha>*

<sha> es un valor hash único que se genera cuando se crea la pila. Por ejemplo, ApiGatewayDomainNameV2*926eeb5ff1*.

Propiedad referenciable: *<httpapi-LogicalId>*.DomainName

CloudFormation recursos generados cuando **AWS::Serverless::MicrovmImage** se especifica

Cuando especificas un*AWS::Serverless::MicrovmImage*, AWS Serverless Application Model (AWS SAM) genera un CloudFormation recurso *AWS::Lambda::MicrovmImage* base.

AWS::Lambda::MicrovmImage

LogicalId: *<microvmimage-LogicalId>*

Propiedad referenciable: N/A (debe usar la *LogicalId* para hacer referencia a este CloudFormation recurso)

Además de este CloudFormation recurso, cuando `AWS::Serverless::MicrovmImage` se especifica, AWS SAM también genera CloudFormation recursos para los siguientes escenarios:

Escenarios

- [BuildRoleArn la propiedad no está especificada](#)

BuildRoleArn la propiedad no está especificada

Cuando no especificas la `BuildRoleArn` propiedad de un `AWS::Serverless::MicrovmImage`, AWS SAM genera un `AWS::IAM::Role` CloudFormation recurso con una política en línea que concede los siguientes permisos:

- `s3:GetObject`— con el alcance del bucket de Amazon S3 desde `CodeUri`
- `logs:CreateLogGroup`, `logs:CreateLogStream`, `logs:PutLogEvents` — para registros de compilación

La política de confianza de roles permite que el director del `lambda.amazonaws.com` servicio asuma el rol.

AWS::IAM::Role

LogicalId: *<microvmimage-LogicalId>*BuildRole

CloudFormation recursos generados cuando AWS::Serverless::NetworkConnector se especifica

Al especificar un `AWS::Serverless::NetworkConnector`, AWS Serverless Application Model (AWS SAM) genera un CloudFormation recurso `AWS::Lambda::NetworkConnector` base.

AWS::Lambda::NetworkConnector

LogicalId: *<networkconnector-LogicalId>*

Propiedad referenciable: N/A (debe usar la *LogicalId* para hacer referencia a este CloudFormation recurso)

Además de este CloudFormation recurso, cuando `AWS::Serverless::NetworkConnector` se especifica, AWS SAM también genera CloudFormation recursos para los siguientes escenarios:

Escenarios

- [OperatorRole la propiedad no está especificada](#)

OperatorRole la propiedad no está especificada

Cuando no se especifica la `OperatorRole` propiedad de un `AWS::Serverless::NetworkConnector`, AWS SAM genera un `AWS::IAM::Role` CloudFormation recurso con la política `AWSLambdaNetworkConnectorOperatorPolicy` AWS gestionada adjunta.

La política de confianza de roles permite que el director del `lambda.amazonaws.com` servicio asuma el rol.

AWS::IAM::Role

LogicalId: *<networkconnector-LogicalId>*OperatorRole

CloudFormation recursos generados cuando **AWS::Serverless::WebFunction** se especifica

Al especificar un `AWS::Serverless::WebFunction`, AWS Serverless Application Model (AWS SAM) genera los siguientes CloudFormation recursos base.

AWS::Lambda::WebFunction

LogicalId: *<webfunction-LogicalId>*

Propiedad referenciable: N/A (debe usar la *LogicalId* para hacer referencia a este CloudFormation recurso)

AWS::Lambda::WebFunctionRevision

LogicalId: *<webfunction-LogicalId>*Revision

Propiedad referenciable: N/A (debe usar la para hacer referencia LogicalId a este recurso)
CloudFormation

AWS::Lambda::WebFunctionEndpoint

LogicalId: *<webfunction-LogicalId>*Endpoint

Propiedad referenciable: N/A (debe usar la para hacer referencia LogicalId a este recurso)
CloudFormation

Los tres recursos utilizan el mismo nombre de función. Si no especificas la `FunctionName` propiedad, AWS SAM genera un nombre para el formulario *<webfunction-LogicalId>-<stack-id-prefix>*, donde *stack-id-prefix* aparecen los primeros 8 caracteres del identificador único del ID de la pila. AWS SAM acorta el identificador lógico si es necesario para que el nombre tenga un máximo de 64 caracteres.

Además de estos CloudFormation recursos, cuando `AWS::Serverless::WebFunction` se especifica, AWS SAM también genera CloudFormation recursos para los siguientes escenarios:

Escenarios

- [ExecutionRoleArn la propiedad no está especificada](#)

ExecutionRoleArn la propiedad no está especificada

Cuando no se especifica la `ExecutionRoleArn` propiedad de un `AWS::Serverless::WebFunction`, AWS SAM genera un `AWS::IAM::Role` CloudFormation recurso con la política `AWSLambdaBasicExecutionRole` AWS gestionada adjunta.

La política de confianza de roles permite que el director del `lambda.amazonaws.com` servicio asuma el rol.

AWS::IAM::Role

LogicalId: *<webfunction-LogicalId>*Role

CloudFormation recursos generados cuando AWS::Serverless::WebSocketApi se especifica

Cuando `AWS::Serverless::WebSocketApi` se especifica un, AWS Serverless Application Model (AWS SAM) genera un CloudFormation recurso `AWS::ApiGatewayV2::Api` base. Además, siempre genera un `AWS::ApiGatewayV2::Stage` recurso y, para cada ruta definida en la `Routes` propiedad `AWS::ApiGatewayV2::Route` `AWS::ApiGatewayV2::Integration`, AWS SAM genera y `AWS::Lambda::Permission` recursos.

AWS::ApiGatewayV2::Api

LogicalId: `<websocketapi-LogicalId>`

Propiedad referenciable: N/A (debe usar la `LogicalId` para hacer referencia a este CloudFormation recurso)

AWS::ApiGatewayV2::Stage

LogicalId: `<websocketapi-LogicalId><stage-name>Stage`

`<stage-name>` es la cadena en la que está establecida la propiedad `StageName`. Si no `StageName` se especifica, el valor predeterminado es. `default` Por ejemplo, si establece `StageName` como `prod`, entonces `LogicalId` es `MyWebSocketApiprodStage`.

Propiedad referenciable: `<websocketapi-LogicalId>.Stage`

AWS::ApiGatewayV2::Route

LogicalId: `<websocketapi-LogicalId><normalized-route-key>Route`

`<normalized-route-key>` es la clave de ruta sin caracteres especiales. Por ejemplo, en el caso de la clave de ruta `$connect`, el `LogicalId` es `MyWebSocketApiconnectRoute`.

Propiedad referenciable: N/A (debe usar la `LogicalId` para hacer referencia a este CloudFormation recurso)

AWS::ApiGatewayV2::Integration

LogicalId: `<websocketapi-LogicalId><normalized-route-key>Integration`

Propiedad referenciable: N/A (debe usar la `LogicalId` para hacer referencia a este recurso CloudFormation)

AWS::Lambda::Permission

LogicalId: *<websocketapi-LogicalId><normalized-route-key>*Permission

Propiedad referenciable: N/A (debe usar la para hacer referencia LogicalId a este recurso)
CloudFormation

Además de estos CloudFormation recursos, cuando AWS::Serverless::WebSocketApi se especifica, AWS SAM también genera CloudFormation recursos para los siguientes escenarios:

Escenarios

- [La propiedad de autenticación se AuthType especifica con el valor CUSTOM](#)
- [DomainName se especifica la propiedad](#)
- [BasePath se especifica la propiedad](#)
- [Se ha especificado la propiedad Route53](#)

La propiedad de autenticación se AuthType especifica con el valor CUSTOM

Cuando la Auth propiedad de an AWS::Serverless::WebSocketApi se especifica con el valor AuthType establecido enCUSTOM, AWS SAM genera un AWS::ApiGatewayV2::Authorizer CloudFormation recurso. Si no InvokeRole se especifica, AWS SAM también genera un AWS::Lambda::Permission recurso para permitir que API Gateway invoque la función de autorización.

AWS::ApiGatewayV2::Authorizer

LogicalId: *<websocketapi-LogicalId>*Authorizer

Propiedad referenciable: N/A (debe usar la para hacer referencia LogicalId a este recurso)
CloudFormation

AWS::Lambda::Permission(cuando no InvokeRole se especifica)

LogicalId: *<websocketapi-LogicalId>*AuthorizerPermission

Propiedad referenciable: N/A (debe usar la LogicalId para hacer referencia a este CloudFormation recurso)

DomainName se especifica la propiedad

Cuando se especifica la DomainName Domain propiedad de AWS::Serverless::WebSocketApi una, AWS SAM genera el AWS::ApiGatewayV2::DomainName CloudFormation recurso.

AWS::ApiGatewayV2::DomainName

LogicalId: ApiGatewayDomainNameV2<sha>

<sha> es un valor hash único que se genera cuando se crea la pila. Por ejemplo:
ApiGatewayDomainNameV2926eeb5ff1.

Propiedad referenciable: N/A (debe usar la LogicalId para hacer referencia a este CloudFormation recurso)

BasePath se especifica la propiedad

Cuando AWS::Serverless::WebSocketApi se especifica la Domain propiedad de una, AWS SAM genera AWS::ApiGatewayV2::ApiMapping CloudFormation recursos, uno para cada ruta base especificada. BasePath

AWS::ApiGatewayV2::ApiMapping

LogicalId: <websocketapi-LogicalId><basepath>ApiMapping

<basepath> es el valor de la ruta base. Por ejemplo, si se especifica v1 como ruta base, el LogicalId es MyWebSocketApiv1ApiMapping.

Propiedad referenciable: N/A (debe usar la LogicalId para hacer referencia a este CloudFormation recurso)

Se ha especificado la propiedad Route53

Cuando se especifica la Route53 Domain propiedad de AWS::Serverless::WebSocketApi una, AWS SAM genera un AWS::Route53::RecordSetGroup CloudFormation recurso.

AWS::Route53::RecordSetGroup

LogicalId: RecordSetGroup<sha>

<sha> es un valor hash único que se genera cuando se crea la pila. Por ejemplo:
RecordSetGroup*926eeb5ff1*.

Propiedad referenciable: N/A (debe usar la `LogicalId` para hacer referencia a este CloudFormation recurso)

CloudFormation recursos generados cuando AWS::Serverless::LayerVersion se especifica

Cuando `AWS::Serverless::LayerVersion` se especifica un, AWS Serverless Application Model (AWS SAM) genera un CloudFormation recurso `AWS::Lambda::LayerVersion` base.

AWS::Lambda::LayerVersion

LogicalId: <layerversion-LogicalId>

Propiedad referenciable: N/A (debe usar la `LogicalId` para hacer referencia a este CloudFormation recurso)

CloudFormation recursos generados cuando AWS::Serverless::SimpleTable se especifica

Cuando `AWS::Serverless::SimpleTable` se especifica un, AWS Serverless Application Model (AWS SAM) genera un CloudFormation recurso `AWS::DynamoDB::Table` base.

AWS::DynamoDB::Table

LogicalId: <simpletable-LogicalId>

Propiedad referenciable: N/A (debe usar la `LogicalId` para hacer referencia a este CloudFormation recurso)

CloudFormation recursos generados cuando AWS::Serverless::StateMachine se especifica

Cuando se especifica `AWS::Serverless::StateMachine`, AWS Serverless Application Model (AWS SAM) genera un recurso `AWS::StepFunctions::StateMachine` de base CloudFormation .

AWS::StepFunctions::StateMachine

LogicalId: *<statemachine-LogicalId>*

Propiedad referenciable: N/A (debe usar la *LogicalId* para hacer referencia a este CloudFormation recurso)

Además de este CloudFormation recurso, cuando *AWS::Serverless::StateMachine* se especifica, AWS SAM también genera CloudFormation recursos para los siguientes escenarios:

Escenarios

- [La propiedad del rol no está especificada](#)
- [Se ha especificado un origen de eventos de API](#)
- [Se ha especificado una origen de eventos de puente de eventos \(o bus de eventos\)](#)

La propiedad del rol no está especificada

Cuando no *AWS::Serverless::StateMachine* se especifica la *Role* propiedad de un, AWS SAM genera un *AWS::IAM::Role* CloudFormation recurso.

AWS::IAM::Role

LogicalId: *<statemachine-LogicalId>Role*

Propiedad referenciable: N/A (debe usar la *LogicalId* para hacer referencia a este CloudFormation recurso)

Se ha especificado un origen de eventos de API

Cuando la *Event* propiedad de an *AWS::Serverless::StateMachine* se establece en *Api*, pero no se especifica, se AWS SAM genera el *AWS::ApiGateway::RestApi* CloudFormation recurso. *RestApiId*

AWS::ApiGateway::RestApi

LogicalId: *ServerlessRestApi*

Propiedad referenciable: N/A (debe usar la *LogicalId* para hacer referencia a este CloudFormation recurso)

Se ha especificado una origen de eventos de puente de eventos (o bus de eventos)

Cuando la Event propiedad de an `AWS::Serverless::StateMachine` se establece en uno de los tipos de puente de eventos (o bus de eventos), AWS SAM genera el `AWS::Events::Rule` CloudFormation recurso. Esto se aplica a los siguientes tipos: `EventBridgeRule`, `Schedule`, y `CloudWatchEvents`.

AWS::Events::Rule

LogicalId: `<statemachine-LogicalId><event-LogicalId>`

Propiedad referenciable: N/A (debe usar la `LogicalId` para hacer referencia a este CloudFormation recurso)

Atributos de recursos compatibles con AWS SAM

Los atributos de los recursos son atributos que se pueden añadir AWS SAM y CloudFormation recursos para controlar otros comportamientos y relaciones. Para obtener más información sobre los atributos de los recursos, consulta la [Referencia a los atributos de los recursos](#) en la Guía del usuario de AWS CloudFormation .

AWS SAM admiten un subconjunto de atributos de recursos definidos por CloudFormation. De los atributos de recursos admitidos, algunos se copian únicamente en el CloudFormation recurso generado base del AWS SAM recurso correspondiente y otros se copian en todos los CloudFormation recursos generados como resultado del AWS SAM recurso correspondiente. Para obtener más información sobre CloudFormation los recursos generados a partir de AWS SAM los recursos correspondientes, consulte [Generado CloudFormation recursos para AWS SAM](#).

En la siguiente tabla se resume la compatibilidad de los atributos de los recursos según AWS SAM lo que se indica a [Excepciones](#) continuación.

Atributos de recursos	Recurso(s) generado(s) por el destino
DependsOn	Únicamente el recurso CloudFormation generado por la base. Para obtener información sobre el mapeo entre AWS SAM los recursos y CloudForm
Metadatos ^{1, 2}	ation los recursos base, consulte Generado CloudFormation escenarios de recursos .

Atributos de recursos	Recurso(s) generado(s) por el destino
Condition DeletionPolicy UpdateReplacePolicy	<p>Todos los CloudFormation recursos generados a partir del AWS SAM recurso correspondiente. Para obtener información sobre los escenarios de CloudFormation los recursos generados, consulte Generado CloudFormation escenarios de recursos.</p>

Notas:

1. Para obtener más información sobre el uso del atributo de recurso de Metadata con el tipo de recurso `AWS::Serverless::Function`, consulta [Creación de funciones de eventos de Lambda con tiempos de ejecución personalizados en AWS SAM](#).
2. Para obtener más información sobre el uso del atributo de recurso de Metadata con el tipo de recurso `AWS::Serverless::LayerVersion`, consulta [Creación de capas Lambda en AWS SAM](#).

Excepciones

Hay varias excepciones a las reglas de atributos de recursos descritas anteriormente:

- Para `AWS::Lambda::LayerVersion`, el campo AWS SAM personalizado exclusivo `RetentionPolicy` establece el campo `DeletionPolicy` para los CloudFormation recursos generados. Esto tiene una prevalencia mayor que `DeletionPolicy`. Si no se establece en ningún caso, `DeletionPolicy` se establece de forma predeterminada en `Retain`.
- Para `AWS::Lambda::Version`, si no se especifica la `DeletionPolicy`, el valor predeterminado es `Retain`.
- En el escenario en el que `DeploymentPreferences` se especifica una función sin servidor, los atributos de los recursos no se copian en los siguientes recursos generados CloudFormation :
 - `AWS::CodeDeploy::Application`
 - `AWS::CodeDeploy::DeploymentGroup`
 - El `AWS::IAM::Role` llamado `CodeDeployServiceRole` que se crea para este escenario

- Si la AWS SAM plantilla contiene varias funciones con fuentes de eventos de API que se crean implícitamente, las funciones compartirán el recurso generado `AWS::ApiGateway::RestApi`. En este escenario, si las funciones tienen atributos de recurso diferentes, entonces, para el `AWS::ApiGateway::RestApi` recurso generado, AWS SAM copia los atributos del recurso de acuerdo con las siguientes listas priorizadas:
 - `UpdateReplacePolicy`:
 1. Retain
 2. Snapshot
 3. Delete
 - `DeletionPolicy`:
 1. Retain
 2. Delete

Extensiones de API Gateway para AWS SAM

Las extensiones API Gateway, diseñadas específicamente para AWS, proporcionan personalizaciones y funcionalidades adicionales para el diseño y la administración APIs. Las extensiones de API Gateway son extensiones de la especificación OpenAPI que admiten la autorización AWS específica y las integraciones de API específicas de API Gateway. Para obtener más información acerca de las extensiones de API Gateway, consulte [Extensiones de OpenAPI para API Gateway](#).

AWS SAM admite un subconjunto de extensiones de API Gateway. Para ver qué extensiones de API Gateway son compatibles AWS SAM, consulte la siguiente tabla.

Extensión API Gateway	Compatible con AWS SAM
x-amazon-apigateway-any-objeto de método	Sí
x-amazon-apigateway-apiPropiedad -key-source	No
x-amazon-apigateway-auth Objeto	Sí
x-amazon-apigateway-authorizer Objeto	Sí

<u>x-amazon-apigateway-authtype Propiedad</u>	Sí
<u>x-amazon-apigateway-binaryPropiedad -media-types</u>	Sí
<u>x-amazon-apigateway-documentation Objeto</u>	No
<u>x-amazon-apigateway-endpoint-Objeto de configuración</u>	No
<u>x-amazon-apigateway-gatewayObjeto -responses</u>	Sí
<u>x-amazon-apigateway-gatewayObjeto -responses.GatewayResponse</u>	Sí
<u>x-amazon-apigateway-gatewayObjeto -Responses.responseParameters</u>	Sí
<u>x-amazon-apigateway-gateway-Objeto Responses.responseTemplates</u>	Sí
<u>x-amazon-apigateway-integration Objeto</u>	Sí
<u>x-amazon-apigateway-integrationObjeto .requestTemplates</u>	Sí
<u>x-amazon-apigateway-integrationObjeto .requestParameters</u>	No
<u>x-amazon-apigateway-integrationObjeto .responses</u>	Sí
<u>x-amazon-apigateway-integrationObjeto .response</u>	Sí
<u>x-amazon-apigateway-integrationObjeto.responseTemplates</u>	Sí
<u>x-amazon-apigateway-integrationObjeto .responseParameters</u>	Sí
<u>x-amazon-apigateway-request-propiedad validator</u>	No
<u>x-amazon-apigateway-request-Objeto validators</u>	No
<u>x-amazon-apigateway-requestObjeto -Validators.requestValidator</u>	No

Funciones intrínsecas para AWS SAM

Las funciones intrínsecas son funciones integradas que permiten asignar valores a propiedades que solo están disponibles en tiempo de ejecución. AWS SAM tiene un soporte limitado para determinadas propiedades de las funciones intrínsecas, por lo que no puede resolver

algunas funciones intrínsecas. Por lo tanto, recomendamos agregar la transformación `AWS::LanguageExtensions` para resolver este problema. `AWS::LanguageExtensions` se trata de una macro alojada por la CloudFormation que se pueden utilizar funciones intrínsecas y otras funcionalidades que, por defecto, no están incluidas CloudFormation.

Transform:

- `AWS::LanguageExtensions`
- `AWS::Serverless-2016-10-31`

Note

Si utiliza funciones intrínsecas en la `CodeUri` propiedad, no AWS SAM podrá analizar correctamente los valores. Considere la posibilidad de usar la transformación `AWS::LanguageExtensions` en su lugar.

Para obtener más información, consulte la [sección Propiedades de AWS::Serverless::Function](#).

Para obtener detalles sobre cómo AWS SAMCLI gestiona la `AWS::LanguageExtensions` transformación, incluido el uso de funciones intrínsecas `CodeUri` y otras propiedades empaquetables, consulte. [CloudFormation compatibilidad con extensiones de idioma](#)

Para obtener más información sobre las funciones intrínsecas, consulta la [Referencia de función intrínseca](#) en la Guía del usuario de AWS CloudFormation .

CloudFormation compatibilidad con extensiones de idioma

AWS SAMCLI admite plantillas que utilizan la `AWS::LanguageExtensions` transformación, incluidas `Fn::ForEach`, `Fn::Length`, `Fn::ToJsonString`, y `Fn::FindInMap` con `DefaultValue`. Para obtener información de referencia completa sobre estas construcciones, consulte la [AWS::LanguageExtensions transformación](#) en la Guía del AWS CloudFormation usuario.

Para usar extensiones de idioma en una AWS SAM plantilla, enumere `AWS::LanguageExtensions` en la Transform sección anterior `AWS::Serverless-2016-10-31`:

Transform:

- `AWS::LanguageExtensions`

- AWS::Serverless-2016-10-31

Cuando se AWS SAMCLI detecta `AWS::LanguageExtensions` en la Transform sección de una plantilla y se ha optado por el procesamiento local, se expanden las construcciones de extensiones de idioma de forma local antes de ejecutar las AWS SAM transformaciones. Esto permite `sam build`, `sam package`, `sam deploy`, `sam sync`, `sam validate`, `sam local invoke`, `sam local start-api`, y `sam local start-lambda` trabajar con plantillas que utilizan estas construcciones.

El procesamiento local está desactivado de forma predeterminada. Cuando está desactivada, AWS SAMCLI transfiere la plantilla sin cambios y CloudFormation procesa la `AWS::LanguageExtensions` transformación del lado del servidor en el momento de la implementación. Para conocer los métodos de activación, consulte [Habilitar las extensiones de idioma](#).

La expansión se produce en dos fases cuando está habilitada:

1. Fase 1 (extensiones de idioma): los `Fn::ForEach` bucles se expanden, las funciones intrínsecas se resuelven siempre que es posible y la plantilla se convierte en estándar CloudFormation.
2. Fase 2 (AWS SAM transformación): el AWS SAM traductor procesa la plantilla expandida como de costumbre.

La plantilla original (`Fn::ForEach` intacta) se conserva para su CloudFormation implementación, ya que CloudFormation procesa la `AWS::LanguageExtensions` transformación en el servidor.

Habilitar las extensiones de idioma

El procesamiento local de `AWS::LanguageExtensions` es opcional para cada comando. Hay tres métodos de activación equivalentes, enumerados por orden de prioridad:

1. CLI bandera: transmite una `--language-extensions` sola invocación:

```
sam build --language-extensions
sam package --language-extensions ...
sam deploy --language-extensions ...
```

`--no-language-extensions` deshabilita explícitamente, anulando tanto `samconfig.toml` la variable de entorno como la que se describe a continuación.

2. **samconfig.toml**— conservar la elección por proyecto:

```
[default.build.parameters]
language_extensions = true

[default.package.parameters]
language_extensions = true

[default.deploy.parameters]
language_extensions = true

[default.sync.parameters]
language_extensions = true

[default.local_invoke.parameters]
language_extensions = true

[default.local_start_api.parameters]
language_extensions = true

[default.local_start_lambda.parameters]
language_extensions = true

[default.validate.parameters]
language_extensions = true
```

La `samconfig.toml` entrada se carga de forma predeterminada para el comando, por lo que surte efecto como si se hubiera pasado la bandera y, por lo tanto, prevalece sobre la variable de entorno.

3. Variable de entorno: configurada `SAM_CLI_ENABLE_LANGUAGE_EXTENSIONS=1` para habilitarla en el shell actual:

```
export SAM_CLI_ENABLE_LANGUAGE_EXTENSIONS=1
sam build
sam local invoke MyFunction
```

Los valores verdaderos (no distinguen mayúsculas de minúsculas) son `1`, `true`, `yes`. Todo lo demás, incluida la cadena vacía, se considera desactivado. La variable de entorno solo se consulta cuando ni la CLI marca ni `samconfig.toml` establece un valor.

⚠ Important

Cada comando necesita su propia activación. La transferencia `--language-extensions` a `sam build` no se propaga a otra versión posterior `sam local invoke`: el procesamiento local se decide en función de la invocación del comando. Utilice la variable de entorno o una `samconfig.toml` entrada para activar todos los comandos sin repetir la marca.

Ventilador: ForEach

`Fn::ForEach` genera múltiples recursos, condiciones o resultados a partir de una única definición de plantilla:

```
Transform:
  - AWS::LanguageExtensions
  - AWS::Serverless-2016-10-31

Parameters:
  ServiceNames:
    Type: CommaDelimitedList
    Default: "Users,Orders,Products"

Resources:
  Fn::ForEach::Services:
    - Name
    - !Ref ServiceNames
    - ${Name}Function:
        Type: AWS::Serverless::Function
        Properties:
          Handler: index.handler
          Runtime: python3.12
          CodeUri: ./services/${Name}
```

Al `sam build` ejecutarlo, se expande y `UsersFunction OrdersFunctionProductsFunction`, cada uno se crea a partir de su directorio fuente respectivo.

Propiedades de los artefactos dinámicos

Cuando una propiedad empaquetable utiliza una variable de bucle (por ejemplo, `./services/${Name}`), AWS SAMCLI genera una CloudFormation Mappings sección que asigna cada valor

de colección a su Amazon S3. URI El Fn::ForEach cuerpo se ha reescrito para usarlo de forma que se Fn::FindInMap CloudFormation pueda resolver el artefacto correcto en el momento de la implementación.

Las propiedades de los artefactos reconocidas son las mismas que se reescriben en la actualidad. sam package Esto incluye:

Propiedades de artefactos dinámicos reconocidas por el AWS SAM CLI

Tipo de recurso	Propiedad
AWS::Serverless::Function	CodeUri, ImageUri
AWS::Serverless::LayerVersion	ContentUri
AWS::Serverless::Api	DefinitionUri
AWS::Serverless::HttpApi	DefinitionUri
AWS::Serverless::StateMachine	DefinitionUri
AWS::Serverless::GraphQLApi	SchemaUri , CodeUri
AWS::Serverless::Application	Location
AWS::Lambda::Function	Code, Code.ImageUri
AWS::Lambda::LayerVersion	Content
AWS::ApiGateway::RestApi	BodyS3Location
AWS::ApiGatewayV2::Api	BodyS3Location
AWS::AppSync::GraphQLSchema	DefinitionS3Location
AWS::AppSync::Resolver	RequestMappingTemplateS3Location , ResponseMappingTemplateS3Location , CodeS3Location

Tipo de recurso	Propiedad
<code>AWS::AppSync::FunctionConfiguration</code>	<code>RequestMappingTemplateS3Location</code> , <code>ResponseMappingTemplateS3Location</code> , <code>CodeS3Location</code>
<code>AWS::StepFunctions::StateMachine</code>	<code>DefinitionS3Location</code>
<code>AWS::ElasticBeanstalk::ApplicationVersion</code>	<code>SourceBundle</code>
<code>AWS::Glue::Job</code>	<code>Command.ScriptLocation</code>
<code>AWS::CloudFormation::Stack</code>	<code>TemplateURL</code>
<code>AWS::CloudFormation::StackSet</code>	<code>TemplateURL</code>
<code>AWS::CloudFormation::ModuleVersion</code>	<code>ModulePackage</code>
<code>AWS::CloudFormation::ResourceVersion</code>	<code>SchemaHandlerPackage</code>

En el caso de las propiedades punteadas (por ejemplo, `Command.ScriptLocation` o `Code.ImageUri` activadas `AWS::Glue::Job` o activadas `AWS::Lambda::Function`), el valor se lee y escribe en la ubicación anidada del recurso. El nombre de mapeo generado usa solo el segmento de hoja de la ruta de la propiedad.

Cuando la propiedad tiene una plantilla de bucle, el nombre del mapeo es `SAM<LeafProperty><LoopName>` (por ejemplo, o). `SAMCodeUriServices`
`SAMScriptLocationJobs`

Important

Customer-authored las asignaciones no deben empezar con estos SAM* prefijos, sino que están reservadas para. AWS SAMCLI Consulte [Limitaciones](#).

Por ejemplo, después de: `sam package`

```

Mappings:
  SAMCodeUriServices:
    Users:
      CodeUri: s3://my-bucket/abc123
    Orders:
      CodeUri: s3://my-bucket/def456
    Products:
      CodeUri: s3://my-bucket/ghi789

Resources:
  Fn::ForEach::Services:
    - Name
    - !Ref ServiceNames
    - ${Name}Function:
        Type: AWS::Serverless::Function
        Properties:
          Handler: index.handler
          Runtime: python3.12
          CodeUri: !FindInMap [SAMCodeUriServices, !Ref Name, CodeUri]

```

Múltiples recursos por ForEach organismo

Un solo `Fn::ForEach` cuerpo puede emitir más de un recurso por iteración. Cada recurso se genera para cada valor de colección:

```

Resources:
  Fn::ForEach::Tables:
    - TableName
    - [Users, Orders, Products]
    - ${TableName}Table:
        Type: AWS::DynamoDB::Table
        Properties:
          TableName: !Sub "${AWS::StackName}-${TableName}"
          # ...

  ${TableName}StreamProcessor:
    Type: AWS::Serverless::Function
    Properties:
      CodeUri: stream-processors/${TableName}/
      Events:
        DDBStream:

```

```
Type: DynamoDB
Properties:
  Stream: !GetAtt
    - !Sub "${TableName}Table"
    - StreamArn
```

Mapecto: nombre, resolución de colisiones

Cuando dos recursos del mismo `Fn::ForEach` cuerpo declaran la misma propiedad de artefacto dinámico (por ejemplo, un artefacto dinámico `Api` y un `StateMachine` `useDefinitionUri`), se AWS SAMCLI añade un sufijo tomado de la parte estática de la plantilla de ID lógico del recurso para mantener la exclusividad de los nombres de los mapas:

Ejemplo de mapeo de nombres con sufijos de colisión

Plantilla de recursos	Propiedad	Nombre de mapeo
<code>\${Svc}Api</code>	<code>DefinitionUri</code>	<code>SAMDefinitionUriServicesApi</code>
<code>\${Svc}StateMachine</code>	<code>DefinitionUri</code>	<code>SAMDefinitionUriServicesStateMachine</code>

Cuando no hay colisión, se utiliza el nombre base (por ejemplo, `SAMDefinitionUriServices`).

Parameter-based colecciones

Cuando la `Fn::ForEach` colección es una referencia de parámetros (por ejemplo `!Ref ServiceNames`) y el cuerpo del bucle utiliza una propiedad de artefacto dinámico (por ejemplo, `CodeUri: ./services/${Name}`), AWS SAMCLI necesita los valores de la colección para generar SAM* las asignaciones descritas en. [Propiedades de los artefactos dinámicos](#) Los resuelve cuando procesa la plantilla, a partir de:

1. `--parameter-overrides` pasados al AWS SAMCLI comando.
2. El `Default` valor del parámetro en la plantilla.

Important

Como SAM* las asignaciones se incorporan al empaquetar, debe volver a empaquetarlas siempre que cambie el valor del parámetro (por ejemplo, al añadir un nuevo servicio) para

que las asignaciones incluyan entradas para los nuevos valores. Esto solo se aplica cuando el parámetro genera un bucle de artefactos dinámico; otras anulaciones de parámetros se pueden cambiar en el momento de la implementación, como de costumbre.

```
# Package with the values you intend to deploy with
sam package --language-extensions --parameter-overrides
  ServiceNames="Users,Orders,Products"

# Deploy with the same values
sam deploy --language-extensions --parameter-overrides
  ServiceNames="Users,Orders,Products"
```

Pilas anidadas

`Fn::ForEach` en pilas anidadas, se admiten plantillas (`AWS::CloudFormation::Stack`). AWS SAM CLI pasa la `Parameters` propiedad de la pila principal a la expansión de la plantilla secundaria, de modo que `Fn::ForEach` las colecciones secundarias que hacen referencia a los parámetros proporcionados por la matriz se resuelvan correctamente.

```
# parent.yaml
Resources:
  ChildStack:
    Type: AWS::CloudFormation::Stack
    Properties:
      TemplateURL: ./child.yaml
      Parameters:
        ServiceNames: "Users,Orders,Products"
```

Ventilador anidado: ForEach

Se admiten hasta 5 niveles de anidación, igualando el límite: `CloudFormation`

```
Resources:
  Fn::ForEach::Envs:
    - Env
    - [Dev, Staging, Prod]
    - Fn::ForEach::Services:
      - Svc
      - [Users, Orders]
      - ${Env}${Svc}Function:
```

```

Type: AWS::Serverless::Function
Properties:
  CodeUri: ./services/${Svc}
  Environment:
    Variables:
      STAGE: ${Env}

```

ForEach en salidas

`Fn::ForEach` los bloques también se expanden dentro de la `Outputs` sección, por lo que puedes emitir una salida por cada valor de colección:

```

Outputs:
  Fn::ForEach::FunctionArns:
    - Name
    - [alpha, beta]
    - ${Name}FunctionArn:
        Value: !GetAtt
          - !Sub "${Name}Function"
          - Arn

```

Condiciones y DependsOn

Los recursos emitidos por `Fn::ForEach` pueden transportar `Condition` y `DependsOn` como cualquier otro recurso. La condición o dependencia se replica en cada recurso generado:

```

Conditions:
  IsProd: !Equals [!Ref Environment, prod]

Resources:
  SharedTable:
    Type: AWS::DynamoDB::Table
    # ...

  Fn::ForEach::Functions:
    - Name
    - [api, worker]
    - ${Name}Function:
        Type: AWS::Serverless::Function
        Condition: IsProd
        DependsOn: SharedTable
        Properties:

```

```
Handler: main.handler
CodeUri: functions/${Name}/
```

&Sintaxis {identifier}

La `&{identifier}` sintaxis elimina los caracteres no alfanuméricos del valor sustituido, lo que resulta útil para generar identificadores lógicos válidos a partir de valores como las direcciones IP:

```
Fn::ForEach::Hosts:
- IP
- ["10.0.0.1", "10.0.0.2"]
- Host&{IP}:
  Type: AWS::EC2::Instance
  # Expands to Host10001, Host10002
```

Funciones intrínsecas compatibles

Las siguientes funciones intrínsecas se resuelven localmente durante la expansión:

Las funciones intrínsecas se resolvieron localmente durante la expansión de las extensiones de lenguaje

Función	Description (Descripción)
<code>Fn::ForEach</code>	Expansión de bucles.
<code>Fn::Length</code>	Devuelve el recuento de elementos de la lista.
<code>Fn::ToJsonString</code>	Convierte un valor en una JSON cadena.
<code>Fn::FindInMap</code>	Búsqueda de mapas, incluida la opcional <code>DefaultValue</code> .
<code>Fn::If</code>	Selección de valores condicionales.
<code>Fn::Sub</code>	Sustitución de cadenas.
<code>Fn::Join</code>	Concatenación de cadenas.
<code>Fn::Split</code>	División de cadenas.
<code>Fn::Select</code>	Selección de elementos de lista.

Función	Description (Descripción)
<code>Fn::Base64</code>	Codificación Base64.
<code>Fn::Equals</code> , <code>Fn::And</code> , <code>Fn::Or</code> , <code>Fn::Not</code>	Evaluación del estado.
Ref	Referencias de parámetros y pseudoparámetros.

Las funciones que requieren recursos desplegados (`Fn::GetAtt`, `Fn::ImportValue`, `Fn::GetAZs`) se conservan para que se resuelvan en CloudFormation el momento de la implementación.

Errores de validación

Los siguientes problemas de plantilla se detectan localmente antes de que se ejecute la AWS SAM transformación:

Se producen errores de validación durante la expansión de las extensiones de idioma

Causa	Mensaje de error
El <code>Fn::ForEach</code> valor tiene un formato incorrecto: no es una lista, no tiene exactamente 3 elementos o tiene un identificador de bucle que no es una cadena.	<code>Fn::ForEach::<key> layout is incorrect (planteado como <code>InvalidTemplateException</code>).</code>
<code>Fn::ForEach</code> Están anidados más de 5 niveles.	<code>Fn::ForEach nesting depth of <N> exceeds the maximum allowed depth of 5. CloudFormation supports up to 5 nested Fn::ForEach loops.</code>
La colección se resuelve en una lista vacía (por ejemplo, un <code>CommaDelimitedList</code> parámetro con <code>Default: ""</code>).	Sin error: el ciclo se omite silenciosamente y no se emite ningún recurso.

Causa	Mensaje de error
El !Ref elemento de la colección apunta a un parámetro que no está declarado en la plantilla.	Sin error: la referencia no resuelta se conserva en la plantilla. En el momento de la implementación, lo CloudFormation resolverá del lado del servidor.

Limitaciones

- Las colecciones deben poder resolverse en el momento. `build/package Fn::ForEach` Las colecciones que utilizan `Fn::GetAtt` referencias dinámicas de SSM/Secrets Manager o Manager no se pueden expandir localmente. `Fn::ImportValue` En su `--parameter-overrides` lugar, utilice un parámetro con.
- Los mapeos dinámicos de artefactos se fijan en el momento del paquete. Cuando una `Fn::ForEach` colección es una referencia de parámetro y el cuerpo del bucle utiliza una propiedad de artefacto dinámico (por ejemplo, `CodeUri: ./services/${Name}`), las SAM* asignaciones generadas solo contienen entradas para los valores de los parámetros que se resolvieron en el momento del paquete. Si `--parameter-overrides` cambias ese parámetro en el momento del despliegue sin volver a empaquetarlo, los nuevos valores no tendrán entradas de mapeo y el despliegue fallará. Esto no se aplica a los parámetros que no se utilizan para controlar un artefacto `Fn::ForEach` dinámico.
- DeletionPolicy** y **UpdateReplacePolicy** se validan y resuelven durante la expansión. Son compatibles con `Ref` los parámetros, pero no con otras funciones intrínsecas.
- Límite de anidación. Se `Fn::ForEach` pueden anidar hasta 5 niveles de, lo que coincide con el límite del lado del CloudFormation servidor.
- Nombres de mapeo reservados. Los nombres de mapeo que comiencen con alguno de los siguientes están reservados para el AWS SAMCLI; no cree sus propios mapeos con estos prefijos:
 - `SAMCodeUri`, `SAMImageUri`, `SAMContentUri`, `SAMDefinitionUri`, `SAMSchemaUri`, `SAMBodyS3Location`, `SAMDefinitionS3Location`, `SAMTemplateURL`, `SAMCode`, `SAMContent` — emitido por `sam package` para propiedades de artefactos dinámicos. Consulte la [Propiedades de los artefactos dinámicos](#) tabla.
 - `SAMLayers`— emitido `sam build` cuando una función `Fn::ForEach` generada recoge referencias de capa de dependencia generadas automáticamente (Lambda coloca las compilaciones en capas AWS SAMCLI en una pila anidada). Este prefijo no tiene ninguna propiedad correspondiente creada por el usuario; se añade automáticamente.

Telemetría

AWS SAMCLI emite un evento de *CFNLanguageExtensions* telemetría cuando se invoca un comando con `--language-extensions` (o su equivalente en una variable de entorno) y la plantilla declara la transformación. `AWS::LanguageExtensions` El evento se activa una vez por invocación y no se transmite el contenido de la plantilla. Cuando el procesamiento local está desactivado (opción predeterminada), no se desencadena ningún evento.

Ejemplos de aplicaciones sin servidor para AWS SAM

En esta sección se incluyen dos aplicaciones de muestra: una que procesa eventos de DynamoDB y otra que procesa eventos de Amazon S3. Cada ejemplo explica el step-by-step proceso de creación de una aplicación. Además, ambos incluyen detalles sobre la configuración de los orígenes de eventos y recursos de AWS. Ambos comienzan por identificar lo que se debe hacer antes de empezar y siguen con los pasos para inicializar, probar, empaquetar e implementar la aplicación.

Temas

- [Procese eventos de DynamoDB con AWS SAM](#)
- [Procese eventos de Amazon S3 con AWS SAM](#)

Procese eventos de DynamoDB con AWS SAM

Con esta aplicación de ejemplo, se basa en lo aprendido en la información general y en la guía de inicio rápido, e instala otra aplicación de ejemplo. Esta aplicación consta de una función de eventos de Lambda que invoca una fuente de eventos de tabla de DynamoDB. La función de Lambda es muy sencilla: registra los datos que se han transferido a través del mensaje de origen del evento.

En este ejercicio se muestra cómo imitar los mensajes de origen de eventos que se pasan a las funciones de Lambda cuando se invocan.

Antes de empezar

Asegúrate de que ha realizado la configuración requerida en el [Instala la AWS SAM CLI](#).

Paso 1: Inicializar la aplicación

En esta sección, se descarga el paquete de la aplicación, que consta de una AWS SAM plantilla y un código de aplicación.

Para inicializar la aplicación

1. Ejecuta el siguiente comando en un AWS SAM CLI indicador de comandos.

```
sam init \  
--location gh:aws-samples/cookiecutter-aws-sam-dynamodb-python \
```

```
--no-input
```

Tenga `gh:` en cuenta que en el comando anterior se expande a la GitHub URL `https://github.com/`.

2. Revisa el contenido del directorio creado por el comando (`dynamodb_event_reader/`):
 - `template.yaml`— Define dos AWS recursos que necesita la aplicación Read DynamoDB: una función Lambda y una tabla de DynamoDB. La plantilla también define el mapeo entre los dos recursos.
 - Directorio de `read_dynamodb_event/`: contiene el código de la aplicación DynamoDB.

Paso 2: probar la aplicación localmente

Para realizar pruebas locales, utiliza el CLI de la AWS SAM para generar un ejemplo de evento de DynamoDB e invocar la función de Lambda:

```
sam local generate-event dynamodb update | sam local invoke --event - ReadDynamoDBEvent
```

El `generate-event` comando crea un mensaje fuente de eventos de prueba, como los mensajes que se crean cuando todos los componentes se implementan en la nube. AWS Este mensaje de origen de eventos se canaliza a la función `ReadDynamoDBEvent` Lambda.

Comprueba que los mensajes esperados se impriman en la consola, en función del código fuente en `app.py`.

Paso 3: crear el paquete de la aplicación

Tras probar la aplicación de forma local, se utiliza AWS SAMCLI para crear un paquete de despliegue, que se utiliza para desplegar la aplicación en la AWS nube.

Creación del paquete de implementación de Lambda

1. Crea un bucket de S3 en la ubicación donde desea guardar el código empaquetado. Si quieres utilizar un bucket de S3 existente, omite este paso.

```
aws s3 mb s3://bucketname
```

2. Crea el paquete de implementación ejecutando el siguiente comando `package` CLI en el indicador de comandos.

```
sam package \  
  --template-file template.yaml \  
  --output-template-file packaged.yaml \  
  --s3-bucket bucketname
```

El nuevo archivo de plantilla, `packaged.yaml`, se especifica al implementar la aplicación en el siguiente paso.

Paso 4: implementación de la aplicación

Ahora que ha creado el paquete de despliegue, lo utiliza para desplegar la aplicación AWS en la nube. A continuación, pruebe la aplicación.

Para implementar la aplicación sin servidor en AWS Cloud

- En AWS SAMCLI, utilice el comando `deploy CLI` para implementar todos los recursos que definió en la plantilla.

```
sam deploy \  
  --template-file packaged.yaml \  
  --stack-name sam-app \  
  --capabilities CAPABILITY_IAM \  
  --region us-east-1
```

En el comando, el parámetro `--capabilities` permite a AWS CloudFormation crear un rol de IAM.

CloudFormation crea los AWS recursos que se definen en la plantilla. Puede acceder a los nombres de estos recursos en la CloudFormation consola.

Para probar la aplicación sin servidor en el AWS Cloud

1. Abre la consola de DynamoDB.
2. Inserte un registro en la tabla que acaba de crear.
3. Ve a la pestaña Métricas de la tabla y selecciona Ver todas las CloudWatch métricas. En la CloudWatch consola, selecciona Registros para poder ver la salida del registro.

Siguientes pasos

Para seguir informándote sobre AWS SAM ellos, consulta los siguientes recursos:

- [Serverless Land](#): sitio que reúne patrones sin servidor, ejemplos de código y recursos de aprendizaje específicos para aplicaciones AWS SAM.

Procese eventos de Amazon S3 con AWS SAM

Con esta aplicación de muestra, puedes basarte en lo aprendido en los ejemplos anteriores e instalar una aplicación más compleja. Esta aplicación consta de una función de eventos de Lambda que es invocada por una fuente de eventos de carga de objetos de Amazon S3. En este ejercicio se muestra cómo acceder a AWS los recursos y realizar llamadas AWS de servicio mediante una función Lambda.

Esta aplicación de muestra, que no tiene servidor, procesa eventos de creación de objetos en Amazon S3. Para cada imagen que se carga en un bucket, Amazon S3 detecta el evento creado por el objeto e invoca una función de Lambda. La función de Lambda invoca a Amazon Rekognition para detectar el texto de la imagen. A continuación, almacena los resultados devueltos por Amazon Rekognition en una tabla de DynamoDB.

Note

Con esta aplicación de ejemplo, los pasos se realizan en un orden ligeramente diferente al de los ejemplos anteriores. El motivo es que este ejemplo requiere que se creen AWS los recursos y que se configuren los permisos de IAM antes de poder probar la función Lambda de forma local. Vamos a aprovechar CloudFormation para crear los recursos y configurar los permisos por usted. De lo contrario, tendrás que hacerlo manualmente antes de poder probar la función de Lambda de forma local.

Como este ejemplo es más complicado, asegúrate primero de estar familiarizado con la instalación de las aplicaciones del ejemplo anterior antes de ejecutar esta.

Antes de empezar

Asegúrate de que has completado la configuración requerida en el [Instala la AWS SAM CLI](#).

Paso 1: Inicializar la aplicación

En esta sección, descargará la aplicación de muestra, que consta de una AWS SAM plantilla y un código de aplicación.

Para inicializar la aplicación

1. Ejecuta el siguiente comando en un AWS SAM CLI indicador de comandos.

```
sam init \
--location https://github.com/aws-samples/cookiecutter-aws-sam-s3-rekognition-
dynamodb-python \
--no-input
```

2. Revisa el contenido del directorio creado por el comando (`aws_sam_ocr/`):

- `template.yaml`— Define tres AWS recursos que necesita la aplicación Amazon S3: una función Lambda, un bucket de Amazon S3 y una tabla de DynamoDB. La plantilla también define el mapeo y los permisos entre estos recursos.
- Directorio de `src/`: contiene el código de la aplicación de Amazon S3.
- `SampleEvent.json`: El origen del evento de muestra, que se utiliza para las pruebas locales.

Paso 2: Crear el paquete de la aplicación

Antes de poder probar esta aplicación de forma local, debe utilizarla AWS SAMCLI para crear un paquete de implementación, que utilizará para implementar la aplicación en la nube. AWS Esta implementación crea los AWS recursos y permisos necesarios para probar la aplicación a nivel local.

Creación del paquete de implementación de Lambda

1. Crea un bucket de S3 en la ubicación donde desea guardar el código empaquetado. Si quieres utilizar un bucket de S3 existente, omite este paso.

```
aws s3 mb s3://bucketname
```

2. Crea el paquete de implementación ejecutando el siguiente comando package CLI en el indicador de comandos.


```
sam package \  
  --template-file template.yaml \  
  --output-template-file packaged.yaml \  
  --s3-bucket bucketname
```

El nuevo archivo de plantilla, `packaged.yaml`, se especifica al implementar la aplicación en el siguiente paso.

Paso 3: implementar de la aplicación

Ahora que ha creado el paquete de implementación, lo usa para implementar la aplicación AWS en la nube. A continuación, pruebe la aplicación invocándola en la AWS nube.

Para implementar la aplicación sin servidor en AWS Cloud

- En el AWS SAMCLI, utilice el `deploy` comando para implementar todos los recursos que definió en la plantilla.

```
sam deploy \  
  --template-file packaged.yaml \  
  --stack-name aws-sam-ocr \  
  --capabilities CAPABILITY_IAM \  
  --region us-east-1
```

En el comando, el `--capabilities` parámetro permite CloudFormation crear un rol de IAM.

CloudFormation crea los AWS recursos que se definen en la plantilla. Puede acceder a los nombres de estos recursos en la CloudFormation consola.

Para probar la aplicación sin servidor en AWS Cloud

1. Cargua una imagen en el bucket de Amazon S3 que creaste para esta aplicación de muestra.
2. Abre la consola de DynamoDB y busca la tabla que se ha creado. Consulta la tabla para ver los resultados devueltos por Amazon Rekognition.
3. Comprueba que la tabla de DynamoDB contenga nuevos registros, que a su vez contengan el texto que Amazon Rekognition encontró en la imagen cargada.

Paso 4: probar la aplicación de forma local

Antes de poder probar la aplicación de forma local, primero debe recuperar los nombres de los AWS recursos creados por CloudFormation.

- Recupere el nombre de la clave y el nombre del bucket de Amazon S3 de CloudFormation. Modifica el archivo `SampleEvent.json` sustituyendo los valores claves del objeto, el nombre del bucket y el ARN del bucket.
- Recupera el nombre de la tabla de DynamoDB. Este nombre se utiliza para el siguiente comando `sam local invoke`.

Utiliza la CLI de AWS SAM para generar un ejemplo de evento de Amazon S3 e invocar la función de Lambda:

```
TABLE_NAME=Table name obtained from CloudFormation console sam local invoke --event  
SampleEvent.json
```

La parte `TABLE_NAME=` establece el nombre de la tabla de DynamoDB. El parámetro `--event` especifica el archivo que contiene el mensaje del evento de prueba que se va a pasar a la función de Lambda.

Ahora puedes comprobar que se crearon los registros de DynamoDB esperados, en función de los resultados devueltos por Amazon Rekognition.

Próximos pasos a seguir

Para seguir informándote sobre AWS SAM ellos, consulta los siguientes recursos:

- [Serverless Land](#): sitio que reúne patrones sin servidor, ejemplos de código y recursos de aprendizaje específicos para aplicaciones AWS SAM.

Desarrolle su aplicación sin servidor con AWS SAM

Esta sección contiene temas sobre la validación de la AWS SAM plantilla y la creación de la aplicación con dependencias. También contiene temas sobre su uso AWS SAM en determinados casos de uso, como el trabajo con capas Lambda, el uso de aplicaciones anidadas, el control del acceso a API APIs Gateway, la organización de los recursos con AWS Step Functions y la firma de código de las aplicaciones. Los tres hitos principales que debe completar para desarrollar su aplicación se enumeran a continuación.

Temas

- [Crea tu aplicación en AWS SAM](#)
- [Defina su infraestructura con AWS SAM](#)
- [Cree su aplicación con AWS SAM](#)

Crea tu aplicación en AWS SAM

Después de completar [Introducción](#) y de leer [Cómo usar AWS Serverless Application Model \(AWS SAM\)](#), estará listo para crear un proyecto de AWS SAM en su entorno de desarrollador. Su AWS SAM proyecto servirá como punto de partida para escribir su aplicación sin servidor. Para obtener una lista de las opciones de AWS SAMCLI `sam init` comando, consulte [sam init](#).

El AWS Serverless Application Model comando Command Line Interface (AWS SAMCLI) `sam init` proporciona opciones para inicializar una nueva aplicación sin servidor que consta de:

- Una AWS SAM plantilla para definir el código de infraestructura.
- Una estructura de carpetas que organiza la aplicación.
- Configuración para sus funciones de AWS Lambda eventos.

Para crear un AWS SAM proyecto, consulte los temas de estas secciones.

Note

`sam init` incluye plantillas de proyectos para funciones duraderas en tiempos de ejecución compatibles (PythonTypeScript, Java). Estas plantillas proporcionan código de inicio y configuración para crear aplicaciones sin servidor con estado.

Temas

- [Inicialización de aplicación sin servidor](#)
- [Opciones para el inicio de sesión](#)
- [Resolución de problemas](#)
- [Ejemplos](#)
- [Más información](#)
- [Sigüientes pasos](#)

Inicialización de aplicación sin servidor

Cómo inicializar una nueva aplicación sin servidor mediante la de AWS SAM CLI

1. `cd` a un directorio de inicio.
2. Ejecute lo siguiente en la línea de comandos:

```
$ sam init
```

3. La CLI de AWS SAM lo guiará a través de un flujo interactivo para crear una nueva aplicación sin servidor.

Note

Como se explica en [Tutorial: Implemente una aplicación Hello World con AWS SAM](#), este comando inicializa la aplicación sin servidor y crea el directorio del proyecto. Este directorio contendrá varios archivos y carpetas. El archivo más importante es `template.yaml`. Esta es tu plantilla. AWS SAM Su versión de python debe coincidir con la versión de python que aparece en el archivo `template.yaml` que creó el comando `sam init`.

Elegir una plantilla inicial

Una plantilla consta de lo siguiente:

1. Una AWS SAM plantilla para su código de infraestructura.

2. Un directorio inicial del proyecto que organiza los archivos del proyecto. Por ejemplo, esto puede incluir:
 - a. Una estructura para el código de la función de Lambda y sus dependencias.
 - b. Una carpeta `events` que contiene los eventos de prueba para las pruebas locales.
 - c. Una carpeta `tests` para respaldar las pruebas unitarias.
 - d. Un archivo `samconfig.toml` para configurar los ajustes del proyecto.
 - e. Un archivo `ReadMe` y otros archivos básicos de inicio del proyecto.

A continuación, se muestra un ejemplo de un directorio inicial para un proyecto:

```
sam-app
### README.md
### __init__.py
### events
#   ### event.json
### hello_world
#   ### __init__.py
#   ### app.py
#   ### requirements.txt
### samconfig.toml
### template.yaml
### tests
    ### __init__.py
    ### integration
    #   ### __init__.py
    #   ### test_api_gateway.py
    ### requirements.txt
    ### unit
        ### __init__.py
        ### test_handler.py
```

Puede seleccionar de una lista de plantillas de inicio rápido de AWS disponibles o proporcionar su propia ubicación de plantilla personalizada.

Para elegir una AWS Plantilla de inicio rápido

1. Cuando se le solicite, seleccione Plantillas de inicio rápido de AWS .
2. Seleccione una plantilla de inicio AWS rápido para empezar. A continuación, se muestra un ejemplo:

```
Which template source would you like to use?
```

- 1 - AWS Quick Start Templates
- 2 - Custom Template Location

```
Choice: 1
```

```
Choose an AWS Quick Start application template
```

- 1 - Hello World Example
- 2 - Multi-step workflow
- 3 - Serverless API
- 4 - Scheduled task
- 5 - Standalone function
- 6 - Data processing
- 7 - Hello World Example With Powertools
- 8 - Infrastructure event management
- 9 - Serverless Connector Hello World Example
- 10 - Multi-step workflow with Connectors
- 11 - Lambda EFS example
- 12 - DynamoDB Example
- 13 - Machine Learning

```
Template: 4
```

Para elegir su propia ubicación de plantilla personalizada

1. Cuando se le solicite, seleccione la ubicación de la plantilla personalizada.

```
Which template source would you like to use?
```

- 1 - AWS Quick Start Templates
- 2 - Custom Template Location

```
Choice: 2
```

2. La CLI de AWS SAM le pedirá que proporcione una ubicación para la plantilla.

```
Template location (git, mercurial, http(s), zip, path):
```

Proporcione cualquiera de las siguientes ubicaciones al archivo.zip de la plantilla:

- Repositorio GitHub: la ruta al archivo.zip de su repositorio GitHub. El archivo debe estar en la raíz del repositorio.

- Repositorio Mercurial: la ruta al archivo.zip de su repositorio Mercurial. El archivo debe estar en la raíz del repositorio.
 - Ruta.zip: una ruta HTTPS o local a su archivo.zip.
3. La CLI de AWS SAM inicializará la aplicación sin servidor con su plantilla personalizada.

Elija un tiempo de ejecución

Al elegir una plantilla de inicio rápido de AWS , la CLI de AWS SAM le pide que seleccione un tiempo de ejecución para las funciones de Lambda. La lista de opciones que muestra la CLI de AWS SAM son los tiempos de ejecución compatibles de forma nativa con Lambda.

- El [tiempo de ejecución](#) proporciona un entorno específico del lenguaje que se ejecuta en el entorno de ejecución.
- [Cuando se implementa en Nube de AWS, el servicio Lambda invoca la función en un entorno de ejecución.](#)

Puedes utilizar cualquier otro lenguaje de programación con un tiempo de ejecución personalizado. Para ello, debe crear manualmente la estructura inicial de la aplicación. A continuación, puede utilizar `sam init` para inicializar rápidamente la aplicación configurando una ubicación de plantilla personalizada.

A partir de tu selección, la CLI de AWS SAM crea el directorio de inicio para el código y las dependencias de la función de Lambda.

Si Lambda admite varios administradores de dependencias para su tiempo de ejecución, se le solicitará que elija el administrador de dependencias que prefiera.

Elija un tipo de paquete

Cuando elige una plantilla de inicio rápido de AWS y un tiempo de ejecución, la CLI de AWS SAM le pide que seleccione un tipo de paquete. El tipo de paquete determina cómo se implementan las funciones de Lambda para usarlas con el servicio de Lambda. Los dos tipos de paquetes compatibles son:

1. Imagen del contenedor: contiene el sistema operativo base, el tiempo de ejecución, las extensiones de Lambda, el código de tu aplicación y sus dependencias.
2. Un archivo de archivo .zip, que incluye el código de la aplicación y sus dependencias.

Para obtener más información sobre paquetes de implementación, consulta [Paquetes de implementación de Lambda](#) en la Guía para desarrolladores de AWS Lambda .

A continuación, se muestra un ejemplo de estructura de directorios de una aplicación con una función de Lambda empaquetada como imagen de contenedor. AWS SAMCLIDescarga la imagen y crea un directorio Dockerfile en el directorio de la función para especificar la imagen.

```
sam-app
### README.md
### __init__.py
### events
#   ### event.json
### hello_world
#   ### Dockerfile
#   ### __init__.py
#   ### app.py
#   ### requirements.txt
### samconfig.toml
### template.yaml
### tests
    ### __init__.py
    ### unit
        ### __init__.py
        ### test_handler.py
```

A continuación, se muestra un ejemplo de estructura de directorios de una aplicación con una función empaquetada como un archivo de archivos.zip.

```
sam-app
### README.md
### __init__.py
### events
#   ### event.json
### hello_world
#   ### __init__.py
#   ### app.py
#   ### requirements.txt
### samconfig.toml
### template.yaml
### tests
    ### __init__.py
    ### integration
```



```
#   ## __init__.py
#   ## test_api_gateway.py
## requirements.txt
## unit
    ## __init__.py
    ## test_handler.py
```

Configuración AWS X-Ray tracing

Puede optar por activar el AWS X-Ray rastreo. Para obtener más información, consulta [¿Qué es? AWS X-Ray](#) en la Guía para AWS X-Ray desarrolladores.

Si la activas, AWS SAMCLI configura tu AWS SAM plantilla. A continuación, se muestra un ejemplo:

```
Globals:
  Function:
    ...
    Tracing: Active
  Api:
    TracingEnabled: True
```

Configure la supervisión con Amazon CloudWatch Application Insights

Puede optar por activar la supervisión mediante Amazon CloudWatch Application Insights. Para obtener más información, consulte [Amazon CloudWatch Application Insights](#) en la Guía del CloudWatch usuario de Amazon.

Si la activas, AWS SAMCLI configurará tu AWS SAM plantilla. A continuación, se muestra un ejemplo:

```
Resources:
  ApplicationResourceGroup:
    Type: AWS::ResourceGroups::Group
    Properties:
      Name:
        Fn::Join:
          - ''
          - - ApplicationInsights-SAM-
            - Ref: AWS::StackName
      ResourceQuery:
        Type: CLOUDFORMATION_STACK_1_0
```

```
ApplicationInsightsMonitoring:
  Type: AWS::ApplicationInsights::Application
  Properties:
    ResourceGroupName:
      Fn::Join:
        - ''
        - - ApplicationInsights-SAM-
          - Ref: AWS::StackName
    AutoConfigurationEnabled: 'true'
    DependsOn: ApplicationResourceGroup
```

Asigne un nombre a la aplicación.

Indique un nombre para su solicitud. La CLI de AWS SAM crea una carpeta de nivel superior para la aplicación con este nombre.

Opciones para el inicio de sesión

Estas son algunas de las opciones principales que puede utilizar con el comando `sam init`. Para obtener una lista de todas las opciones, consulta [sam init](#).

Inicializa una aplicación mediante una ubicación de plantilla personalizada

Utiliza la opción `--location` y proporciona una ubicación de plantilla personalizada compatible. A continuación, se muestra un ejemplo:

```
$ sam init --location https://github.com/aws-samples/sessions-with-aws-sam/raw/master/
starter-templates/web-app.zip
```

Inicializa una aplicación sin el flujo interactivo

Utiliza la opción `--no-interactive` y proporciona las opciones de configuración en la línea de comandos para omitir el flujo interactivo. A continuación, se muestra un ejemplo:

```
$ sam init --no-interactive --runtime go1.x --name go-demo --dependency-manager mod --
app-template hello-world
```

Resolución de problemas

Para solucionar el problema AWS SAMCLI, consulte. [AWS SAM CLI solución de problemas](#)

Ejemplos

Inicialice una nueva aplicación sin servidor mediante Hello World AWS Plantilla de inicio

Para ver este ejemplo, consulta [Paso 1: inicializar la aplicación de muestra Hello World](#) en el tutorial: Implementación de una aplicación de Hello World.

Inicializa una nueva aplicación sin servidor con una ubicación de plantilla personalizada

Los siguientes son ejemplos de cómo proporcionar una ubicación GitHub a la plantilla personalizada:

```
$ sam init --location gh:aws-samples/cookiecutter-aws-sam-python
$ sam init --location git+sh://git@github.com/aws-samples/cookiecutter-aws-sam-python.git
$ sam init --location hg+ssh://hg@bitbucket.org/repo/template-name
```

A continuación, se muestra un ejemplo de una ruta de archivo local:

```
$ sam init --location /path/to/template.zip
```

A continuación, se muestra un ejemplo de una ruta a la que se puede acceder mediante HTTPS:

```
$ sam init --location https://github.com/aws-samples/sessions-with-aws-sam/raw/master/starter-templates/web-app.zip
```

Más información

Para obtener más información sobre el uso del comando `sam init`, consulta los enlaces siguientes:

- [Aprendizaje AWS SAM: sam init](#) — serie Serverless Land «Aprendizaje AWS SAM» sobre YouTube.
- [Estructuración de aplicaciones sin servidor para su uso con la CLI de AWS SAM\(Sesiones con SAM S2E7\)](#): Sesiones con la serie AWS SAM en YouTube.

Siguientes pasos

Ahora que ha creado su AWS SAM proyecto, está listo para empezar a crear su aplicación. Consulte [Defina su infraestructura con AWS SAM](#) para obtener instrucciones detalladas sobre las tareas que debe completar para hacerlo.

Defina su infraestructura con AWS SAM

Ahora que ha creado su proyecto, está listo para definir su infraestructura de aplicaciones con él AWS SAM. Para ello, configure la AWS SAM plantilla para definir los recursos y las propiedades de la aplicación, que es el `template.yaml` archivo AWS SAM del proyecto.

Los temas de esta sección proporcionan contenido sobre cómo definir la infraestructura en la AWS SAM plantilla (el `template.yaml` archivo). También contiene temas sobre la definición de recursos para casos de uso específicos, como el trabajo con capas Lambda, el uso de aplicaciones anidadas, el control del acceso a API Gateway, la organización de los recursos con AWS Step Functions APIs, la firma de código de las aplicaciones y la validación de la plantilla. AWS SAM

Temas

- [Defina los recursos de la aplicación en su AWS SAM plantilla](#)
- [Configure y administre el acceso a los recursos en su AWS SAM plantilla](#)
- [Controle el acceso a la API con su AWS SAM plantilla](#)
- [Aumente la eficiencia con las capas Lambda con AWS SAM](#)
- [Reutilice el código y los recursos mediante aplicaciones anidadas en AWS SAM](#)
- [Gestione eventos basados en el tiempo con el EventBridge programador en AWS SAM](#)
- [Orquestando AWS SAM recursos con AWS Step Functions](#)
- [Configura la firma de código para tu AWS SAM aplicación](#)
- [Validar archivos AWS SAM de plantilla](#)

Defina los recursos de la aplicación en su AWS SAM plantilla

Los AWS recursos que utiliza la aplicación sin servidor se definen en la `Resources` sección de la AWS SAM plantilla. Al definir un recurso, identifica qué es el recurso, cómo interactúa con otros recursos y cómo se puede acceder a él (es decir, los permisos del recurso).

La **Resources** sección de la AWS SAM plantilla puede contener una combinación de CloudFormation recursos y AWS SAM recursos. Además, puedes usar AWS SAM la sintaxis abreviada para los siguientes recursos:

AWS SAM sintaxis abreviada	Qué hace con un recurso relacionado AWS
<u>AWS::Serverless::API</u>	Crea una colección de recursos y métodos de Amazon API Gateway que se pueden invocar a través de puntos de conexión HTTPS.
<u>AWS::Serverless::Application</u>	Integrar una aplicación sin servidor desde <u>AWS Serverless Application Repository</u> o desde un bucket de Amazon S3 como una aplicación anidada.
<u>AWS::Serverless::Connector</u>	Configura permisos entre dos recursos. Para obtener una introducción a los conectores, consulta <u>Administrar los permisos de recursos con AWS SAM conectores</u> .
<u>AWS::Serverless::Function</u>	Crea una función de AWS Lambda evento, un rol de ejecución AWS Identity and Access Management (IAM) y mapeos de fuentes de eventos que activan la función.
<u>AWS::Serverless::API GraphQL</u>	crea y configura una AWS AppSync GraphQL API para su aplicación sin servidor.
<u>AWS::Serverless::HttpApi</u>	Crea una API HTTP de Amazon API Gateway, que permite crear API RESTful con menor latencia y costos que las API de REST.
<u>AWS::Serverless::LayerVersion</u>	Crea una Lambda LayerVersion que contiene el código de biblioteca o de tiempo de ejecución que necesita una función Lambda.
<u>AWS::Serverless::SimpleTable</u>	Crea una tabla de DynamoDB con una claves principal de atributo único.

AWS SAM sintaxis abreviada	Qué hace con un recurso relacionado AWS
<code>AWS::Serverless::StateMachine</code>	Crea una máquina de AWS Step Functions estados que se puede utilizar para organizar AWS Lambda funciones y otros AWS recursos a fin de crear flujos de trabajo complejos y sólidos.

Los recursos anteriores también se enumeran en [AWS SAM recursos y propiedades](#).

Para obtener información de referencia sobre todos los tipos de AWS recursos y propiedades CloudFormation y el AWS SAM soporte, consulte la [referencia sobre tipos de AWS recursos y propiedades](#) en la Guía del AWS CloudFormation usuario.

Configure y administre el acceso a los recursos en su AWS SAM plantilla

Para que AWS los recursos interactúen entre sí, se deben configurar el acceso y los permisos adecuados entre los recursos. Para ello, es necesario configurar los usuarios, las funciones y las políticas AWS Identity and Access Management (de IAM) para que la interacción se lleve a cabo de forma segura.

Todos los temas de esta sección están relacionados con la configuración del acceso a los recursos definidos en la plantilla. Esta sección comienza con las prácticas recomendadas generales. En los dos temas siguientes se analizan dos opciones de las que dispone para configurar el acceso y los permisos entre los recursos a los que se hace referencia en su aplicación sin servidor: los AWS SAM conectores y las plantillas de AWS SAM políticas. El último tema proporciona detalles sobre la administración del acceso de los usuarios mediante la misma mecánica que se CloudFormation utiliza para la administración de los usuarios.

Para obtener más información, consulta [Control de acceso con AWS Identity and Access Management](#) en la Guía del usuario de AWS CloudFormation .

El AWS Serverless Application Model (AWS SAM) ofrece dos opciones que simplifican la administración del acceso y los permisos de las aplicaciones sin servidor.

1. AWS SAM conectores
2. Plantillas de política de AWS SAM

AWS SAM conectores

Los conectores son una forma de aprovisionar permisos entre dos recursos. Para ello, describe cómo deben interactuar entre sí en la AWS SAM plantilla. Se pueden definir mediante el atributo de recurso `Connectors` o el tipo de recurso `AWS::Serverless::Connector`. Los conectores permiten el aprovisionamiento `Read` y el `Write` acceso a datos y eventos entre una combinación de AWS recursos. Para obtener más información sobre AWS SAM los conectores, consulte [Administrar los permisos de recursos con AWS SAM conectores](#).

AWS SAM Plantillas de política de

AWS SAM las plantillas de políticas son conjuntos de permisos predefinidos que puede añadir a sus AWS SAM plantillas para gestionar el acceso y los permisos entre sus funciones de AWS Lambda eventos, las máquinas de AWS Step Functions estado y los recursos con los que interactúan. Para obtener más información sobre las plantillas AWS SAM de políticas, consulte [AWS SAM Plantillas de política de](#).

AWS CloudFormation mecanismos

CloudFormation los mecanismos incluyen la configuración de los usuarios, las funciones y las políticas de IAM para gestionar los permisos entre los AWS recursos. Para obtener más información, consulte [Gestión de AWS SAM permisos con CloudFormation mecanismos](#).

Prácticas recomendadas

En todas sus aplicaciones sin servidor, puede utilizar varios métodos para configurar los permisos entre los recursos. Por lo tanto, puede seleccionar la mejor opción para cada escenario y utilizar varias opciones juntas en todas sus aplicaciones. Estas son algunas cosas que se deben tener en cuenta cuando se elige la mejor opción para usted:

- AWS SAM Tanto los conectores como las plantillas de políticas reducen la experiencia en IAM necesaria para facilitar las interacciones seguras entre los recursos. AWS Utiliza conectores y plantillas de políticas cuando sean compatibles.
- AWS SAM Los conectores proporcionan una sintaxis abreviada sencilla e intuitiva para definir los permisos en AWS SAM las plantillas y requieren el mínimo nivel de experiencia en IAM. Si se admiten tanto AWS SAM los conectores como las plantillas de políticas, utilice AWS SAM conectores.
- AWS SAM los conectores pueden aprovisionar datos `Read` y eventos y `Write` acceder a ellos entre los recursos de AWS SAM origen y destino compatibles. Para obtener una lista de recursos

admitidos, consulta [AWS SAM referencia de conector](#). Cuando sea compatible, utilice AWS SAM conectores.

- Si bien las plantillas de AWS SAM políticas se limitan a los permisos entre las funciones de Lambda, las máquinas de estado de Step Functions y los AWS recursos con los que interactúan, las plantillas de políticas admiten todas las operaciones de CRUD. Cuando sea compatible y cuando haya una plantilla AWS SAM de política disponible para su escenario, utilice plantillas de AWS SAM políticas. Para obtener una lista de las plantillas de políticas disponibles, consulta [AWS SAM Plantillas de política de](#).
- Para todos los demás escenarios, o cuando se requiera granularidad, utilice CloudFormation mecanismos.

Administrar los permisos de recursos con AWS SAM conectores

Los conectores son un tipo de recurso abstracto AWS Serverless Application Model (AWS SAM), identificado como `AWS::Serverless::Connector`, que proporciona permisos simples y con un amplio alcance entre los recursos de las aplicaciones sin servidor.

Ventajas de AWS SAM conectores

Al elaborar automáticamente las políticas de acceso adecuadas entre los recursos, los conectores le permiten crear sus aplicaciones sin servidor y centrarse en la arquitectura de las aplicaciones sin necesidad de conocimientos especializados en capacidades de AWS autorización, lenguaje de políticas y configuraciones de seguridad específicas de los servicios. Por lo tanto, los conectores son una gran ventaja para los desarrolladores que se están iniciando en el desarrollo sin servidores o para los desarrolladores experimentados que desean aumentar su velocidad de desarrollo.

Utilización AWS SAM conectores

Utiliza el atributo `Connectors` de recurso integrándolo en un recurso fuente. A continuación, defina el recurso de destino y describa cómo deben fluir los datos o los eventos entre esos recursos. AWS SAM a continuación, redacta las políticas de acceso necesarias para facilitar las interacciones requeridas.

A continuación se describe cómo se escribe este atributo de recurso:

```
AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31
...
Resources:
```



```

<source-resource-logical-id>:
  Type: <resource-type>
  ...
  Connectors:
    <connector-name>:
      Properties:
        Destination:
          <properties-that-identify-destination-resource>
      Permissions:
        <permission-types-to-provision>
  ...

```

Cómo funcionan los conectores

Note

En esta sección se explica cómo los conectores proporcionan los recursos necesarios entre bastidores. Esto sucede automáticamente al utilizar conectores.

En primer lugar, el atributo Connectors de recurso incrustado se transforma en un tipo de recurso `AWS::Serverless::Connector`. Su ID lógico se crea automáticamente como `<source-resource-logical-id><embedded-connector-logical-id>`.

Por ejemplo, aquí hay un conector integrado:

```

AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31
...
Resources:
  MyFunction:
    Type: AWS::Lambda::Function
    Connectors:
      MyConn:
        Properties:
          Destination:
            Id: MyTable
        Permissions:
          - Read
          - Write
  MyTable:
    Type: AWS::DynamoDB::Table

```

Esto generará el siguiente recurso AWS::Serverless::Connector:

```
Transform: AWS::Serverless-2016-10-31
Resources:
  ...
  MyFunctionMyConn:
    Type: AWS::Serverless::Connector
    Properties:
      Source:
        Id: MyFunction
      Destination:
        Id: MyTable
      Permissions:
        - Read
        - Write
```

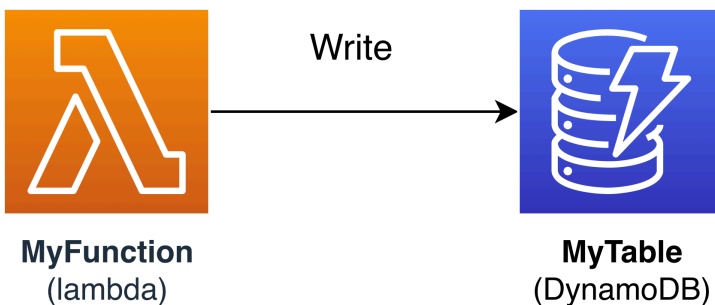
Note

También puede definir los conectores en la AWS SAM plantilla mediante esta sintaxis. Esto se recomienda cuando el recurso de origen está definido en una plantilla independiente de la del conector.

A continuación, se redactan automáticamente las políticas de acceso necesarias para esta conexión. Para obtener más información acerca de los recursos generados por conectores, consulta [CloudFormation recursos generados cuando se especifica AWS::Serverless::Connector](#).

Ejemplo de conectores

El siguiente ejemplo muestra cómo utilizar conectores para escribir datos de una función de AWS Lambda eventos en una tabla de Amazon DynamoDB.



```
Transform: AWS::Serverless-2016-10-31
Resources:
```

```

MyTable:
  Type: AWS::Serverless::SimpleTable
MyFunction:
  Type: AWS::Serverless::Function
  Connectors:
    MyConn:
      Properties:
        Destination:
          Id: MyTable
        Permissions:
          - Write
  Properties:
    Runtime: nodejs16.x
    Handler: index.handler
    InlineCode: |
      const AWS = require("aws-sdk");
      const docClient = new AWS.DynamoDB.DocumentClient();
      exports.handler = async (event, context) => {
        await docClient.put({
          TableName: process.env.TABLE_NAME,
          Item: {
            id: context.awsRequestId,
            event: JSON.stringify(event)
          }
        }).promise();
      }
  Environment:
    Variables:
      TABLE_NAME: !Ref MyTable

```

El atributo de recurso `Connectors` está integrado en el recurso fuente de la función de Lambda. La tabla de DynamoDB se define como el recurso de destino que utiliza la propiedad `Id`. Los conectores proporcionarán permisos `Write` entre estos dos recursos.

Al implementar la AWS SAM plantilla en CloudFormation, AWS SAM redactará automáticamente las políticas de acceso necesarias para que esta conexión funcione.

Recursos de origen y destino compatibles para los conectores

Compatibilidad con conectores `Read` y tipos de permisos de datos y eventos `Write` entre una combinación selecta de conexiones de recursos de origen y destino. Por ejemplo, los conectores admiten una conexión `Write` entre un recurso `AWS::ApiGateway::RestApi` de origen y un recurso `AWS::Lambda::Function` de destino.

Los recursos de origen y destino se pueden definir mediante una combinación de propiedades compatibles. Los requisitos de propiedad dependerán de la conexión que se realice y de dónde estén definidos los recursos.

 Note

Los conectores pueden proporcionar permisos entre los tipos de recursos compatibles sin servidor y con servidor.

Para obtener una lista de conexiones de recursos compatibles y sus requisitos de propiedad, consulta [Tipos de recursos de origen y destino admitidos para los conectores](#).

Defina los permisos de lectura y escritura en AWS SAM

En AWS SAM, Read y Write los permisos se pueden aprovisionar en un solo conector:

```
AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31
...
Resources:
  MyFunction:
    Type: AWS::Lambda::Function
    Connectors:
      MyTableConn:
        Properties:
          Destination:
            Id: MyTable
          Permissions:
            - Read
            - Write
  MyTable:
    Type: AWS::DynamoDB::Table
```

Para obtener más información sobre el uso de conectores, consulte [AWS SAM referencia de conector](#).

Defina los recursos mediante otras propiedades compatibles en AWS SAM

Para los recursos de origen y destino, cuando se definen en la misma plantilla, utiliza la propiedad Id. Si lo desea, puedes añadirse un Qualifier para reducir el alcance del recurso definido.

Cuando el recurso no esté dentro de la misma plantilla, utiliza una combinación de propiedades admitidas.

- Para obtener una lista de las combinaciones de propiedades compatibles para los recursos de origen y destino, consulta [Tipos de recursos de origen y destino admitidos para los conectores](#).
- Para obtener una descripción de las propiedades que puedes usar con los conectores, consulta [AWS::Serverless::Conector](#).

Cuando define un recurso de origen con una propiedad distinta de Id, utiliza la propiedad SourceReference.

```
AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31
...
Resources:
  <source-resource-logical-id>:
    Type: <resource-type>
    ...
    Connectors:
      <connector-name>:
        Properties:
          SourceReference:
            Qualifier: <optional-qualifier>
            <other-supported-properties>
          Destination:
            <properties-that-identify-destination-resource>
          Permissions:
            <permission-types-to-provision>
```

A continuación, se muestra un ejemplo del uso de un Qualifier para reducir el alcance de un recurso de Amazon API Gateway:

```
AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31
...
Resources:
  MyApi:
    Type: AWS::Serverless::Api
    Connectors:
      ApiToLambdaConn:
        Properties:
```

```
SourceReference:
  Qualifier: Prod/GET/foobar
Destination:
  Id: MyFunction
Permissions:
  - Write
...
```

A continuación, se muestra un ejemplo en el que se utiliza una combinación compatible de Arn y Type para definir un recurso de destino a partir de otra plantilla:

```
AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31
...
Resources:
  MyFunction:
    Type: AWS::Serverless::Function
    Connectors:
      TableConn:
        Properties:
          Destination:
            Type: AWS::DynamoDB::Table
            Arn: !GetAtt MyTable.Arn
...
```

Para obtener más información sobre el uso de conectores, consulte [AWS SAM referencia de conector](#).

Cree varios conectores a partir de una sola fuente en AWS SAM

Dentro de un recurso de origen, puedes definir varios conectores, cada uno con un recurso de destino diferente.

```
AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31
...
Resources:
  MyFunction:
    Type: AWS::Serverless::Function
    Connectors:
      BucketConn:
        Properties:
          Destination:
```

```

        Id: amzn-s3-demo-bucket
    Permissions:
        - Read
        - Write
    SQSConn:
        Properties:
            Destination:
                Id: MyQueue
            Permissions:
                - Read
                - Write
    TableConn:
        Properties:
            Destination:
                Id: MyTable
            Permissions:
                - Read
                - Write
    TableConnWithTableArn:
        Properties:
            Destination:
                Type: AWS::DynamoDB::Table
                Arn: !GetAtt MyTable.Arn
            Permissions:
                - Read
                - Write
    ...

```

Para obtener más información sobre el uso de conectores, consulte [AWS SAM referencia de conector](#).

Cree conectores multidestino en AWS SAM

Dentro de un recurso de origen, puedes definir un conector único con varios recursos de destino. A continuación, se muestra un ejemplo de recurso fuente de una función de Lambda para conectarse a un bucket de Amazon Simple Storage Service (Amazon S3) y a una tabla de DynamoDB:

```

AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31
...
Resources:
    MyFunction:
        Type: AWS::Serverless::Function

```

```
Connectors:
  WriteAccessConn:
    Properties:
      Destination:
        - Id: OutputBucket
        - Id: CredentialTable
      Permissions:
        - Write
...
OutputBucket:
  Type: AWS::S3::Bucket
CredentialTable:
  Type: AWS::DynamoDB::Table
```

Para obtener más información sobre el uso de conectores, consulte [AWS SAM referencia de conector](#).

Defina los atributos de los recursos con los conectores en AWS SAM

Los atributos de los recursos se pueden definir para que los recursos especifiquen comportamientos y relaciones adicionales. Para obtener más información sobre los atributos de los recursos, consulta la [referencia a los atributos de los recursos](#) en la Guía del usuario AWS CloudFormation .

Puedes añadir atributos de recursos al conector integrado definiéndolos al mismo nivel que las propiedades del conector. Cuando la AWS SAM plantilla se transforme en el momento de la implementación, los atributos pasarán a los recursos generados.

```
AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31
...
Resources:
  MyFunction:
    Type: AWS::Serverless::Function
    Connectors:
      MyConn:
        DeletionPolicy: Retain
        DependsOn: AnotherFunction
        Properties:
          ...
```

Para obtener más información sobre el uso de conectores, consulte [AWS SAM referencia de conector](#).

Más información

Para obtener más información sobre el uso de AWS SAM conectores, consulte los siguientes temas:

- [AWS::Serverless: :Conector](#)
- [Defina los permisos de lectura y escritura en AWS SAM](#)
- [Defina los recursos mediante otras propiedades compatibles en AWS SAM](#)
- [Cree varios conectores a partir de una sola fuente en AWS SAM](#)
- [Cree conectores multidestino en AWS SAM](#)
- [Defina los permisos de lectura y escritura en AWS SAM](#)
- [Defina los atributos de los recursos con los conectores en AWS SAM](#)

Proporcionar información

Para enviar comentarios sobre los conectores, [envíe un nuevo número al repositorio AWS GitHubserverless-application-model](#).

AWS SAM Plantillas de política de

El AWS Serverless Application Model (AWS SAM) le permite elegir de una lista de plantillas de políticas para limitar los permisos de sus funciones de eventos y máquinas de AWS Step Functions estado de Lambda a los recursos que utiliza su aplicación.

AWS SAM las aplicaciones AWS Serverless Application Repository que utilizan plantillas de políticas no requieren ningún reconocimiento especial del cliente para implementar la aplicación desde. AWS Serverless Application Repository

Si desea solicitar que se agregue una nueva plantilla de política, haga lo siguiente:

1. Envía una solicitud de incorporación de datos utilizando el archivo fuente `policy_templates.json` de la rama del proyecto. `develop` AWS SAM GitHub [Puedes encontrar el archivo fuente en `policy\_templates.json` en el sitio web](#). GitHub
2. Envía una incidencia en el AWS SAM GitHub proyecto que incluya los motivos de tu solicitud de cambios y un enlace a la solicitud. Utiliza este enlace para enviar una nueva edición: [Issues AWS Serverless Application Model](#).

Sintaxis

Para cada plantilla de política que especifiques en tu archivo de AWS SAM plantilla, siempre debes especificar un objeto que contenga los valores de los marcadores de posición de la plantilla de política. Si una plantilla de política no requiere ningún valor de marcador de posición, debe especificar un objeto vacío.

YAML

```
MyFunction:
  Type: AWS::Serverless::Function
  Properties:
    Policies:
      - PolicyTemplateName1:      # Policy template with placeholder value
        Key1: Value1
      - PolicyTemplateName2: {}   # Policy template with no placeholder value
```

Note

Si ha configurado una política de IAM o ha administrado políticas mediante Lambda, la plantilla de política se puede configurar sin utilizar un objeto vacío.

Ejemplos

Ejemplo 1: plantilla de política con valores de marcador de posición

El siguiente ejemplo muestra que la plantilla de política [SQSPollerPolicy](#) espera QueueName como recurso. La AWS SAM plantilla recupera el nombre de la cola "MyQueue" de Amazon SQS, que puede crear en la misma aplicación o solicitarla como parámetro de la aplicación.

```
MyFunction:
  Type: 'AWS::Serverless::Function'
  Properties:
    CodeUri: ${codeuri}
    Handler: hello.handler
    Runtime: python2.7
    Policies:
      - SQSPollerPolicy:
        QueueName:
          !GetAtt MyQueue.QueueName
```

Ejemplo 2: plantilla de política sin valores de marcador de posición

El ejemplo siguiente contiene la plantilla de política [CloudWatchPutMetricPolicy](#), que no tiene valores de marcador de posición.

Note

Aunque no haya valores de marcador de posición, debe especificar un objeto vacío; de lo contrario, se producirá un error.

```
MyFunction:
  Type: 'AWS::Serverless::Function'
  Properties:
    CodeUri: ${codeuri}
    Handler: hello.handler
    Runtime: python2.7
    Policies:
      - CloudWatchPutMetricPolicy: {}
```

Ejemplo 3: plantilla de políticas con valores de marcadores de posición y una política de IAM normal

El siguiente ejemplo contiene la AmazonSQSFullAccess política y [DynamoDBCrudPolicy](#) la plantilla de política. La AmazonSQSFullAccess política es una política de IAM y no una AWS SAM política, por lo que no es necesario especificar un objeto vacío, ya que la política se transferiría directamente al CloudFormation.

```
MyFunction:
  Type: 'AWS::Serverless::Function'
  Properties:
    CodeUri: ${codeuri}
    Handler: hello.handler
    Runtime: python2.7
    Policies:
      - AmazonSQSFullAccess // IAM policy could be set without passing an empty object
      - DynamoDBCrudPolicy: // SAM specific policy, has a defined structure
        TableName:
```

!Ref SampleTable

Tabla de plantillas de políticas

La siguiente es una tabla de las plantillas de políticas disponibles.

Plantilla de política	Description (Descripción)		
AcmGetCertificatePolicy	Otorga permiso para leer un certificado desde AWS Gestor de certificados.		
AMIDescribePolicy	Concede permiso para describir imágenes de máquina de Amazon (AMI).		
AthenaQueryPolicy	Concede permisos para ejecutar consultas de Athena.		
AWSSecretsManagerGetSecretValuePolicy	Concede permiso para obtener el valor secreto del secreto AWS Secrets Manager especificado.		
AWSSecretsManagerRotationPolicy	Concede permiso para transferir un secreto AWS Secrets Manager.		
CloudFormationDescribeStacksPolicy	Da permiso para describir las CloudFormation pilas.		
CloudWatchDashboardPolicy	Otorga permisos para poner las métricas en los CloudWatch paneles de control para que funcionen.		
CloudWatchDescribeAlarmsPolicy	Da permiso para describir el historial CloudWatch de alarmas.		

Plantilla de política	Description (Descripción)		
AlarmHistoryPolicy			
CloudWatchPutMetricPolicy	Da permiso para enviar métricas a CloudWatch.		
CodeCommitCrudPolicy	Otorga permisos a update/delete los objetos create/read/de un CodeCommit repositorio específico.		
CodeCommitReadPolicy	Otorga permisos para leer objetos dentro de un CodeCommit repositorio específico.		
CodePipelineLambdaExecutionPolicy	Da permiso para que una función Lambda invocada por CodePipeline informe del estado del trabajo.		
CodePipelineReadOnlyPolicy	Otorga permiso de lectura para obtener detalles sobre una CodePipeline canalización.		
ComprehendBasicAccessPolicy	Concede permiso para detectar entidades, frases clave, idiomas y sentimientos.		
CostExplorerReadOnlyPolicy	Concede permisos de solo lectura a las API de Cost Explorer de solo lectura para el historial de facturación.		
DynamoDBBackupFullAccessPolicy	Concede permiso de lectura y escritura a las copias de seguridad bajo demanda de DynamoDB para una tabla.		

Plantilla de política	Description (Descripción)		
DynamoDBCrudPolicy	Concede permisos de creación, lectura, actualización y eliminación a una tabla de Amazon DynamoDB.		
DynamoDBReadOnlyPolicy	Concede permiso de solo lectura a una tabla DynamoDB.		
DynamoDBRestorePolicy	Concede permiso para volver a configurar una tabla DynamoDB.		
DynamoDBRestoreFromBackupPolicy	Concede permiso para restaurar una tabla de DynamoDB a partir de una copia de seguridad.		
DynamoDBStreamReadPolicy	Concede permiso para describir y leer flujos y registros de DynamoDB.		
DynamoDBWritePolicy	Concede permiso de solo escritura a una tabla DynamoDB.		
EC2CopyImagePolicy	Concede permiso para copiar imágenes de Amazon EC2.		
EC2DescribePolicy	Concede permiso para describir instancias de Amazon Elastic Compute Cloud (Amazon EC2).		
EcsRunTaskPolicy	Concede permiso para iniciar una nueva tarea para una definición de tarea.		
EFSWriteAccessPolicy	Concede permiso para montar un sistema de archivos de Amazon EFS con acceso de escritura.		

Plantilla de política	Description (Descripción)		
EKSDescribePolicy	Concede permiso para describir o enumerar los clústeres de Amazon EKS.		
ElasticMapReduceJobFlowStepsPolicy	Concede permiso para añadir nuevos pasos a un clúster en ejecución.		
ElasticMapReduceCancelStepsPolicy	Concede permiso para cancelar un paso o pasos pendientes en un clúster en ejecución.		
ElasticMapReduceModifyInstancesFleetPolicy	Concede permiso para enumerar detalles y modificar las capacidades para las flotas de instancia de un clúster.		
ElasticMapReduceModifyInstanceGroupsPolicy	Concede permiso para enumerar los detalles y modificar la configuración de los grupos de instancias de un clúster.		
ElasticMapReduceSetTerminationProtectionPolicy	Concede permiso para establecer la protección de terminación para un clúster.		
ElasticMapReduceTerminateJobFlowsPolicy	Concede permiso para cerrar un clúster.		

Plantilla de política	Description (Descripción)		
ElasticsearchHttpPostPolicy	Otorga permiso POST a Amazon OpenSearch Service.		
EventBridgePutEventsPolicy	Otorga permisos para enviar eventos a EventBridge.		
FilterLogEventsPolicy	Otorga permiso para filtrar CloudWatch los eventos de registros de un grupo de registros especificado.		
FirehoseCreatePutDeletePolicy	Concede permiso para crear, escribir, actualizar y eliminar una secuencia de entrega de Firehose.		
FirehoseWritePolicy	Concede permiso para escribir en una secuencia de entrega de Firehose.		
KinesisCreateDeletePolicy	Concede permiso para crear, publicar y eliminar un flujo de Amazon Kinesis.		
KinesisStreamReadPolicy	Concede permiso para listar y leer un flujo de Amazon Kinesis.		
KMSTDecryptPolicy	Da permiso para descifrar con una clave AWS Key Management Service (AWS KMS).		
KMSEncryptPolicy	Da permiso para cifrar con una clave AWS Key Management Service (AWS KMS).		
LambdaInvokePolicy	Da permiso para invocar una AWS Lambda función, un alias o una versión.		

Plantilla de política	Description (Descripción)		
<u>MobileAnalyticsWriteOnlyAccessPolicy</u>	Concede permiso de solo escritura para poner datos de eventos para todos los recursos de la aplicación.		
<u>OrganizationsListAccountsPolicy</u>	Concede permisos de solo lectura para incluir los nombres e ID de las cuentas secundarias.		
<u>PinpointEndpointAccessPolicy</u>	Concede permiso para obtener y actualizar puntos de conexión para una aplicación de Amazon Pinpoint.		
<u>PollyFullAccessPolicy</u>	Concede permiso de acceso completo a los recursos del léxico de Amazon Polly.		
<u>RekognitionDetectOnlyPolicy</u>	Concede permiso para detectar rostros, etiquetas y texto.		
<u>RekognitionFacesManagementPolicy</u>	Concede permiso para añadir, eliminar y buscar rostros en una colección de Amazon Rekognition.		
<u>RekognitionFacesPolicy</u>	Concede permiso para comparar y detectar rostros y etiquetas.		
<u>RekognitionLabelsPolicy</u>	Concede permiso para detectar etiquetas de objetos y de moderación.		
<u>RekognitionNoDataAccessPolicy</u>	Concede permiso para comparar y detectar rostros y etiquetas.		

Plantilla de política	Description (Descripción)		
<u>RekognitionReadPolicy</u>	Concede permiso para enumerar y buscar rostros.		
<u>RekognitionWriteOnlyAccessPolicy</u>	Concede permiso para crear colecciones e indexar rostros.		
<u>Route53ChangeResourceRecordSetsPolicy</u>	Concede permiso para cambiar los conjuntos de registros de recursos en Route 53.		
<u>S3CrudPolicy</u>	Concede permisos de creación, lectura, actualización y eliminación para actuar en los objetos de un bucket de Amazon S3.		
<u>S3FullAccessPolicy</u>	Concede permiso de acceso completo para actuar sobre los objetos de un bucket de Amazon S3.		
<u>S3ReadPolicy</u>	Concede permiso de solo lectura para leer objetos en un bucket de Amazon Simple Storage Service (Amazon S3).		
<u>S3WritePolicy</u>	Concede permiso de escritura para escribir objetos en un bucket de Amazon S3.		
<u>SageMakerCreateEndpointConfigurationPolicy</u>	Otorga permiso para crear una configuración de punto final en SageMaker AI.		
<u>SageMakerCreateEndpointPolicy</u>	Da permiso para crear un punto final en la SageMaker IA.		

Plantilla de política	Description (Descripción)		
<u>ServerlessRepoReadWriteAccessPolicy</u>	Otorga permiso para crear y enumerar aplicaciones en el AWS Serverless Application Repository servicio.		
<u>SESBulkTemplatedCrudPolicy</u>	Concede permiso para enviar correos electrónicos, correos electrónicos con plantillas, correos electrónicos masivos con plantillas y verificar la identidad.		
<u>SESBulkTemplatedCrudPolicy_v2</u>	Concede permiso para enviar correos electrónicos, correos electrónicos con plantillas y correos electrónicos masivos con plantillas de Amazon SES y para verificar la identidad.		
<u>SESCrudPolicy</u>	Concede permiso para enviar correo electrónico y verificar la identidad.		
<u>SESEmailTemplateCrudPolicy</u>	Concede permiso para crear, obtener, enumerar, actualizar y eliminar plantillas de correo electrónico de Amazon SES.		
<u>SESSendBouncePolicy</u>	Concede SendBounce permiso a una identidad de Amazon Simple Email Service (Amazon SES).		
<u>SNSCrudPolicy</u>	Concede permisos para crear, publicar y suscribirse a temas de Amazon SNS.		
<u>SNSPublishMessagePolicy</u>	Concede permiso para publicar un mensaje en un tema de Amazon Simple Notification Service (Amazon SNS).		

Plantilla de política	Description (Descripción)		
<u>SQSPollerPolicy</u>	Concede permiso para sondear una cola de Amazon Simple Queue Service (Amazon SQS).		
<u>SQSSendMessagePolicy</u>	Concede permisos para enviar mensajes a una cola de Amazon SQS.		
<u>SSMParameterReadPolicy</u>	Concede permiso para acceder a un parámetro de un almacén de parámetros de Amazon EC2 Systems Manager (SSM) para cargar secretos en esta cuenta. Se utiliza cuando el nombre del parámetro no tiene un prefijo de barra inclinada.		
<u>SSMParameterWithSlashPrefixReadPolicy</u>	Concede permiso para acceder a un parámetro de un almacén de parámetros de Amazon EC2 Systems Manager (SSM) para cargar secretos en esta cuenta. Se utiliza cuando el nombre del parámetro tiene un prefijo de barra diagonal.		
<u>StepFunctionsExecutionPolicy</u>	Concede permiso para iniciar la ejecución de una máquina de estado de Step Functions.		
<u>TextractDetectAnalyzePolicy</u>	Concede acceso para detectar y analizar documentos con Amazon Textract.		
<u>TextractGetResultPolicy</u>	Concede acceso a documentos detectados y analizados desde Amazon Textract.		
<u>TextractPolicy</u>	Proporciona acceso completo a Amazon Textract.		

Plantilla de política	Description (Descripción)		
VPCAccess Policy	Permite crear, eliminar, describir y eliminar interfaces de red elásticas.		

Resolución de problemas

Error de la CLI de SAM: «Debe especificar valores de parámetros válidos para la plantilla de política '<>policy-template-name'»

Al ejecutar `sam build`, verá el siguiente error:

```
"Must specify valid parameter values for policy template '<policy-template-name>'"
```

Esto significa que no pasó un objeto vacío al declarar una plantilla de política que no tiene ningún valor de marcador de posición.

Para solucionar este problema, declare la política como se muestra en el siguiente ejemplo para [CloudWatchPutMetricPolicy](#).

```
MyFunction:
  Policies:
    - CloudWatchPutMetricPolicy: {}
```

AWS SAM lista de plantillas de políticas

A continuación se muestran las plantillas de políticas disponibles, junto con los permisos que se aplican a cada una de ellas. AWS Serverless Application Model (AWS SAM) rellena automáticamente los elementos marcadores de posición (como la AWS región y el identificador de cuenta) con la información adecuada.

Temas

- [AcmGetCertificatePolicy](#)
- [AMIDescribePolicy](#)
- [AthenaQueryPolicy](#)

- [AWS Secrets Manager GetSecretValue Policy](#)
- [AWS Secrets Manager Rotation Policy](#)
- [CloudFormation DescribeStacks Policy](#)
- [CloudWatch Dashboard Policy](#)
- [CloudWatch DescribeAlarmHistory Policy](#)
- [CloudWatch PutMetric Policy](#)
- [CodePipeline Lambda Execution Policy](#)
- [CodePipeline Read Only Policy](#)
- [CodeCommit Crud Policy](#)
- [CodeCommit Read Policy](#)
- [Comprehend Basic Access Policy](#)
- [Cost Explorer Read Only Policy](#)
- [DynamoDB Backup Full Access Policy](#)
- [DynamoDB Crud Policy](#)
- [DynamoDB Read Policy](#)
- [DynamoDB Reconfigure Policy](#)
- [DynamoDB Restore From Backup Policy](#)
- [DynamoDB Stream Read Policy](#)
- [DynamoDB Write Policy](#)
- [EC2 Copy Image Policy](#)
- [EC2 Describe Policy](#)
- [ECS Run Task Policy](#)
- [EFS Write Access Policy](#)
- [EKS Describe Policy](#)
- [Elastic Map Reduce Add Job Flow Steps Policy](#)
- [Elastic Map Reduce Cancel Steps Policy](#)
- [Elastic Map Reduce Modify Instance Fleet Policy](#)
- [Elastic Map Reduce Modify Instance Groups Policy](#)
- [Elastic Map Reduce Set Termination Protection Policy](#)
- [Elastic Map Reduce Terminate Job Flows Policy](#)

- [ElasticsearchHttpPostPolicy](#)
- [EventBridgePutEventsPolicy](#)
- [FilterLogEventsPolicy](#)
- [FirehoseCrudPolicy](#)
- [FirehoseWritePolicy](#)
- [KinesisCrudPolicy](#)
- [KinesisStreamReadPolicy](#)
- [KMSDecryptPolicy](#)
- [KMSEncryptPolicy](#)
- [LambdaInvokePolicy](#)
- [MobileAnalyticsWriteOnlyAccessPolicy](#)
- [OrganizationsListAccountsPolicy](#)
- [PinpointEndpointAccessPolicy](#)
- [PollyFullAccessPolicy](#)
- [RekognitionDetectOnlyPolicy](#)
- [RekognitionFacesManagementPolicy](#)
- [RekognitionFacesPolicy](#)
- [RekognitionLabelsPolicy](#)
- [RekognitionNoDataAccessPolicy](#)
- [RekognitionReadPolicy](#)
- [RekognitionWriteOnlyAccessPolicy](#)
- [Route53ChangeResourceRecordSetsPolicy](#)
- [S3CrudPolicy](#)
- [S3FullAccessPolicy](#)
- [S3ReadPolicy](#)
- [S3WritePolicy](#)
- [SageMakerCreateEndpointConfigPolicy](#)
- [SageMakerCreateEndpointPolicy](#)
- [ServerlessRepoReadWriteAccessPolicy](#)
- [SESBulkTemplatedCrudPolicy](#)

- [SESBulkTemplatedCrudPolicy_v2](#)
- [SESCrudPolicy](#)
- [SESEmailTemplateCrudPolicy](#)
- [SESSendBouncePolicy](#)
- [SNSCrudPolicy](#)
- [SNSPublishMessagePolicy](#)
- [SQSPollerPolicy](#)
- [SQSSendMessagePolicy](#)
- [SSMParameterReadPolicy](#)
- [SSMParameterWithSlashPrefixReadPolicy](#)
- [StepFunctionsExecutionPolicy](#)
- [TextractDetectAnalyzePolicy](#)
- [TextractGetResultPolicy](#)
- [TextractPolicy](#)
- [VPCAccessPolicy](#)

AcmGetCertificatePolicy

Otorga permiso para leer un certificado desde. AWS Gestor de certificados

```
"Statement": [  
  {  
    "Effect": "Allow",  
    "Action": [  
      "acm:GetCertificate"  
    ],  
    "Resource": {  
      "Fn::Sub": [  
        "${certificateArn}",  
        {  
          "certificateArn": {  
            "Ref": "CertificateArn"  
          }  
        }  
      ]  
    }  
  }  
]
```



```
]
```

AMIDescribePolicy

Concede permiso para describir imágenes de máquina de Amazon (AMI).

```
"Statement": [  
  {  
    "Effect": "Allow",  
    "Action": [  
      "ec2:DescribeImages"  
    ],  
    "Resource": "*"   
  }  
]
```

AthenaQueryPolicy

Concede permisos para ejecutar consultas de Athena.

```
"Statement": [  
  {  
    "Effect": "Allow",  
    "Action": [  
      "athena:ListWorkGroups",  
      "athena:GetExecutionEngine",  
      "athena:GetExecutionEngines",  
      "athena:GetNamespace",  
      "athena:GetCatalogs",  
      "athena:GetNamespaces",  
      "athena:GetTables",  
      "athena:GetTable"  
    ],  
    "Resource": "*"   
  },  
  {  
    "Effect": "Allow",  
    "Action": [  
      "athena:StartQueryExecution",  
      "athena:GetQueryResults",  
      "athena>DeleteNamedQuery",  
      "athena:GetNamedQuery",  
      "athena:ListQueryExecutions",  
    ]  
  }  
]
```

```

    "athena:StopQueryExecution",
    "athena:GetQueryResultsStream",
    "athena:ListNamedQueries",
    "athena:CreateNamedQuery",
    "athena:GetQueryExecution",
    "athena:BatchGetNamedQuery",
    "athena:BatchGetQueryExecution",
    "athena:GetWorkGroup"
  ],
  "Resource": {
    "Fn::Sub": [
      "arn:${AWS::Partition}:athena:${AWS::Region}:${AWS::AccountId}:workgroup/${workgroupName}",
      {
        "workgroupName": {
          "Ref": "WorkGroupName"
        }
      }
    ]
  }
}
]

```

AWS::SecretsManager::GetSecretValuePolicy

Da permiso para obtener el valor secreto del AWS Secrets Manager secreto especificado.

```

"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "secretsmanager:GetSecretValue"
    ],
    "Resource": {
      "Fn::Sub": [
        "${secretArn}",
        {
          "secretArn": {
            "Ref": "SecretArn"
          }
        }
      ]
    }
  }
]

```

]

AWS Secrets Manager Rotation Policy

Concede permiso para transferir un secreto AWS Secrets Manager.

```
"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "secretsmanager:DescribeSecret",
      "secretsmanager:GetSecretValue",
      "secretsmanager:PutSecretValue",
      "secretsmanager:UpdateSecretVersionStage"
    ],
    "Resource": {
      "Fn::Sub": "arn:${AWS::Partition}:secretsmanager:${AWS::Region}:
${AWS::AccountId}:secret:*"
    },
    "Condition": {
      "StringEquals": {
        "secretsmanager:resource/AllowRotationLambdaArn": {
          "Fn::Sub": [
            "arn:${AWS::Partition}:lambda:${AWS::Region}:${AWS::AccountId}:function:
${functionName}",
            {
              "functionName": {
                "Ref": "FunctionName"
              }
            }
          ]
        }
      }
    }
  },
  {
    "Effect": "Allow",
    "Action": [
      "secretsmanager:GetRandomPassword"
    ],
    "Resource": "*"
  }
]
```

CloudFormationDescribeStacksPolicy

Da permiso para describir las CloudFormation pilas.

```
"Statement": [  
  {  
    "Effect": "Allow",  
    "Action": [  
      "cloudformation:DescribeStacks"  
    ],  
    "Resource": {  
      "Fn::Sub": "arn:${AWS::Partition}:cloudformation:${AWS::Region}:  
${AWS::AccountId}:stack/*"  
    }  
  }  
]
```

CloudWatchDashboardPolicy

Otorga permisos para poner las métricas en los CloudWatch paneles de control para que funcionen.

```
"Statement": [  
  {  
    "Effect": "Allow",  
    "Action": [  
      "cloudwatch:GetDashboard",  
      "cloudwatch:ListDashboards",  
      "cloudwatch:PutDashboard",  
      "cloudwatch:ListMetrics"  
    ],  
    "Resource": "*"   
  }  
]
```

CloudWatchDescribeAlarmHistoryPolicy

Da permiso para describir el historial de CloudWatch alarmas de Amazon.

```
"Statement": [  
  {  
    "Effect": "Allow",  
    "Action": [  
      "cloudwatch:DescribeAlarmHistory"  
    ]  
  }  
]
```

```
    ],  
    "Resource": "*"    
  }  
]
```

CloudWatchPutMetricPolicy

Da permiso para enviar métricas a CloudWatch.

```
"Statement": [  
  {  
    "Effect": "Allow",  
    "Action": [  
      "cloudwatch:PutMetricData"  
    ],  
    "Resource": "*"    
  }  
]
```

CodePipelineLambdaExecutionPolicy

Da permiso para que una función de eventos de Lambda invocada por AWS CodePipeline informe del estado del trabajo.

```
"Statement": [  
  {  
    "Effect": "Allow",  
    "Action": [  
      "codepipeline:PutJobSuccessResult",  
      "codepipeline:PutJobFailureResult"  
    ],  
    "Resource": "*"    
  }  
]
```

CodePipelineReadOnlyPolicy

Otorga permiso de lectura para obtener detalles sobre una CodePipeline canalización.

```
"Statement": [  
  {  
    "Effect": "Allow",  
    "Action": [  

```

```

    "codepipeline:ListPipelineExecutions"
  ],
  "Resource": {
    "Fn::Sub": [
      "arn:${AWS::Partition}:codepipeline:${AWS::Region}:${AWS::AccountId}:
${pipelineName}",
      {
        "pipelineName": {
          "Ref": "PipelineName"
        }
      }
    ]
  }
}
]

```

CodeCommitCrudPolicy

Otorga permisos para crear, leer, actualizar y eliminar objetos dentro de un CodeCommit repositorio específico.

```

"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "codecommit:GitPull",
      "codecommit:GitPush",
      "codecommit:CreateBranch",
      "codecommit>DeleteBranch",
      "codecommit:GetBranch",
      "codecommit:ListBranches",
      "codecommit:MergeBranchesByFastForward",
      "codecommit:MergeBranchesBySquash",
      "codecommit:MergeBranchesByThreeWay",
      "codecommit:UpdateDefaultBranch",
      "codecommit:BatchDescribeMergeConflicts",
      "codecommit>CreateUnreferencedMergeCommit",
      "codecommit:DescribeMergeConflicts",
      "codecommit:GetMergeCommit",
      "codecommit:GetMergeOptions",
      "codecommit:BatchGetPullRequests",
      "codecommit>CreatePullRequest",
      "codecommit:DescribePullRequestEvents",

```

```
"codecommit:GetCommentsForPullRequest",
"codecommit:GetCommitsFromMergeBase",
"codecommit:GetMergeConflicts",
"codecommit:GetPullRequest",
"codecommit:ListPullRequests",
"codecommit:MergePullRequestByFastForward",
"codecommit:MergePullRequestBySquash",
"codecommit:MergePullRequestByThreeWay",
"codecommit:PostCommentForPullRequest",
"codecommit:UpdatePullRequestDescription",
"codecommit:UpdatePullRequestStatus",
"codecommit:UpdatePullRequestTitle",
"codecommit>DeleteFile",
"codecommit:GetBlob",
"codecommit:GetFile",
"codecommit:GetFolder",
"codecommit:PutFile",
"codecommit>DeleteCommentContent",
"codecommit:GetComment",
"codecommit:GetCommentsForComparedCommit",
"codecommit:PostCommentForComparedCommit",
"codecommit:PostCommentReply",
"codecommit:UpdateComment",
"codecommit:BatchGetCommits",
"codecommit>CreateCommit",
"codecommit:GetCommit",
"codecommit:GetCommitHistory",
"codecommit:GetDifferences",
"codecommit:GetObjectIdentifier",
"codecommit:GetReferences",
"codecommit:GetTree",
"codecommit:GetRepository",
"codecommit:UpdateRepositoryDescription",
"codecommit:ListTagsForResource",
"codecommit:TagResource",
"codecommit:UntagResource",
"codecommit:GetRepositoryTriggers",
"codecommit:PutRepositoryTriggers",
"codecommit:TestRepositoryTriggers",
"codecommit:GetBranch",
"codecommit:GetCommit",
"codecommit:UploadArchive",
"codecommit:GetUploadArchiveStatus",
"codecommit:CancelUploadArchive"
```

```

    ],
    "Resource": {
      "Fn::Sub": [
        "arn:${AWS::Partition}:codecommit:${AWS::Region}:${AWS::AccountId}:
${repositoryName}",
        {
          "repositoryName": {
            "Ref": "RepositoryName"
          }
        }
      ]
    }
  }
}
]

```

CodeCommitReadPolicy

Otorga permisos para leer los objetos de un CodeCommit repositorio específico.

```

"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "codecommit:GitPull",
      "codecommit:GetBranch",
      "codecommit:ListBranches",
      "codecommit:BatchDescribeMergeConflicts",
      "codecommit:DescribeMergeConflicts",
      "codecommit:GetMergeCommit",
      "codecommit:GetMergeOptions",
      "codecommit:BatchGetPullRequests",
      "codecommit:DescribePullRequestEvents",
      "codecommit:GetCommentsForPullRequest",
      "codecommit:GetCommitsFromMergeBase",
      "codecommit:GetMergeConflicts",
      "codecommit:GetPullRequest",
      "codecommit:ListPullRequests",
      "codecommit:GetBlob",
      "codecommit:GetFile",
      "codecommit:GetFolder",
      "codecommit:GetComment",
      "codecommit:GetCommentsForComparedCommit",
      "codecommit:BatchGetCommits",
      "codecommit:GetCommit",

```



```

    "codecommit:GetCommitHistory",
    "codecommit:GetDifferences",
    "codecommit:GetObjectIdentifier",
    "codecommit:GetReferences",
    "codecommit:GetTree",
    "codecommit:GetRepository",
    "codecommit:ListTagsForResource",
    "codecommit:GetRepositoryTriggers",
    "codecommit:TestRepositoryTriggers",
    "codecommit:GetBranch",
    "codecommit:GetCommit",
    "codecommit:GetUploadArchiveStatus"
  ],
  "Resource": {
    "Fn::Sub": [
      "arn:${AWS::Partition}:codecommit:${AWS::Region}:${AWS::AccountId}:",
      "${repositoryName}",
      {
        "repositoryName": {
          "Ref": "RepositoryName"
        }
      }
    ]
  }
}
]

```

ComprehendBasicAccessPolicy

Concede permiso para detectar entidades, frases clave, idiomas y sentimientos.

```

"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "comprehend:BatchDetectKeyPhrases",
      "comprehend:DetectDominantLanguage",
      "comprehend:DetectEntities",
      "comprehend:BatchDetectEntities",
      "comprehend:DetectKeyPhrases",
      "comprehend:DetectSentiment",
      "comprehend:BatchDetectDominantLanguage",
      "comprehend:BatchDetectSentiment"
    ]
  },

```

```

    "Resource": "*"
  }
]

```

CostExplorerReadOnlyPolicy

Otorga permisos de solo lectura a las API de solo lectura (Cost AWS Cost Explorer Explorer) para el historial de facturación.

```

"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "ce:GetCostAndUsage",
      "ce:GetDimensionValues",
      "ce:GetReservationCoverage",
      "ce:GetReservationPurchaseRecommendation",
      "ce:GetReservationUtilization",
      "ce:GetTags"
    ],
    "Resource": "*"
  }
]

```

DynamoDBBackupFullAccessPolicy

Concede permiso de lectura y escritura a las copias de seguridad bajo demanda de DynamoDB para una tabla.

```

"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "dynamodb:CreateBackup",
      "dynamodb:DescribeContinuousBackups"
    ],
    "Resource": {
      "Fn::Sub": [
        "arn:${AWS::Partition}:dynamodb:${AWS::Region}:${AWS::AccountId}:table/${tableName}",
        {
          "tableName": {
            "Ref": "TableName"
          }
        ]
      ]
    }
  }
]

```

```

    }
  }
]
}
},
{
  "Effect": "Allow",
  "Action": [
    "dynamodb:DeleteBackup",
    "dynamodb:DescribeBackup",
    "dynamodb:ListBackups"
  ],
  "Resource": {
    "Fn::Sub": [
      "arn:${AWS::Partition}:dynamodb:${AWS::Region}:${AWS::AccountId}:table/
${tableName}/backup/*",
      {
        "tableName": {
          "Ref": "TableName"
        }
      }
    ]
  }
}
}
]

```

DynamoDBCrudPolicy

Concede permisos de creación, lectura, actualización y eliminación a una tabla de Amazon DynamoDB.

```

"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "dynamodb:GetItem",
      "dynamodb:DeleteItem",
      "dynamodb:PutItem",
      "dynamodb:Scan",
      "dynamodb:Query",
      "dynamodb:UpdateItem",
      "dynamodb:BatchWriteItem",
      "dynamodb:BatchGetItem",

```

```

        "dynamodb:DescribeTable",
        "dynamodb:ConditionCheckItem"
    ],
    "Resource": [
        {
            "Fn::Sub": [
                "arn:${AWS::Partition}:dynamodb:${AWS::Region}:${AWS::AccountId}:table/${tableName}",
                {
                    "tableName": {
                        "Ref": "TableName"
                    }
                }
            ]
        },
        {
            "Fn::Sub": [
                "arn:${AWS::Partition}:dynamodb:${AWS::Region}:${AWS::AccountId}:table/${tableName}/index/*",
                {
                    "tableName": {
                        "Ref": "TableName"
                    }
                }
            ]
        }
    ]
}
]

```

DynamoDBReadPolicy

Concede permiso de solo lectura a una tabla DynamoDB.

```

"Statement": [
    {
        "Effect": "Allow",
        "Action": [
            "dynamodb:GetItem",
            "dynamodb:Scan",
            "dynamodb:Query",
            "dynamodb:BatchGetItem",
            "dynamodb:DescribeTable"
        ]
    }
]

```

```

    "Resource": [
      {
        "Fn::Sub": [
          "arn:${AWS::Partition}:dynamodb:${AWS::Region}:${AWS::AccountId}:table/
${tableName}",
          {
            "tableName": {
              "Ref": "TableName"
            }
          }
        ]
      },
      {
        "Fn::Sub": [
          "arn:${AWS::Partition}:dynamodb:${AWS::Region}:${AWS::AccountId}:table/
${tableName}/index/*",
          {
            "tableName": {
              "Ref": "TableName"
            }
          }
        ]
      }
    ]
  }
]

```

DynamoDBReconfigurePolicy

Concede permiso para volver a configurar una tabla DynamoDB.

```

"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "dynamodb:UpdateTable"
    ],
    "Resource": {
      "Fn::Sub": [
        "arn:${AWS::Partition}:dynamodb:${AWS::Region}:${AWS::AccountId}:table/
${tableName}",
        {
          "tableName": {
            "Ref": "TableName"
          }
        ]
      ]
    }
  }
]

```

```

    }
  }
]
}
]

```

DynamoDBRestoreFromBackupPolicy

Concede permiso para restaurar una tabla de DynamoDB a partir de una copia de seguridad.

```

"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "dynamodb:RestoreTableFromBackup"
    ],
    "Resource": {
      "Fn::Sub": [
        "arn:${AWS::Partition}:dynamodb:${AWS::Region}:${AWS::AccountId}:table/${tableName}/backup/*",
        {
          "tableName": {
            "Ref": "TableName"
          }
        }
      ]
    }
  },
  {
    "Effect": "Allow",
    "Action": [
      "dynamodb:PutItem",
      "dynamodb:UpdateItem",
      "dynamodb>DeleteItem",
      "dynamodb:GetItem",
      "dynamodb:Query",
      "dynamodb:Scan",
      "dynamodb:BatchWriteItem"
    ],
    "Resource": {
      "Fn::Sub": [
        "arn:${AWS::Partition}:dynamodb:${AWS::Region}:${AWS::AccountId}:table/${tableName}",

```

```

        {
            "tableName": {
                "Ref": "TableName"
            }
        }
    ]
}
]

```

DynamoDBStreamReadPolicy

Concede permiso para describir y leer flujos y registros de DynamoDB.

```

"Statement": [
    {
        "Effect": "Allow",
        "Action": [
            "dynamodb:DescribeStream",
            "dynamodb:GetRecords",
            "dynamodb:GetShardIterator"
        ],
        "Resource": {
            "Fn::Sub": [
                "arn:${AWS::Partition}:dynamodb:${AWS::Region}:${AWS::AccountId}:table/${tableName}/stream/${streamName}",
                {
                    "tableName": {
                        "Ref": "TableName"
                    },
                    "streamName": {
                        "Ref": "StreamName"
                    }
                }
            ]
        }
    },
    {
        "Effect": "Allow",
        "Action": [
            "dynamodb:ListStreams"
        ],
        "Resource": {
            "Fn::Sub": [

```

```

    "arn:${AWS::Partition}:dynamodb:${AWS::Region}:${AWS::AccountId}:table/
    ${tableName}/stream/*",
    {
      "tableName": {
        "Ref": "TableName"
      }
    }
  ]
}
]

```

DynamoDBWritePolicy

Concede permiso de solo escritura a una tabla DynamoDB.

```

"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "dynamodb:PutItem",
      "dynamodb:UpdateItem",
      "dynamodb:BatchWriteItem"
    ],
    "Resource": [
      {
        "Fn::Sub": [
          "arn:${AWS::Partition}:dynamodb:${AWS::Region}:${AWS::AccountId}:table/
          ${tableName}",
          {
            "tableName": {
              "Ref": "TableName"
            }
          }
        ]
      },
      {
        "Fn::Sub": [
          "arn:${AWS::Partition}:dynamodb:${AWS::Region}:${AWS::AccountId}:table/
          ${tableName}/index/*",
          {
            "tableName": {
              "Ref": "TableName"
            }
          }
        ]
      }
    ]
  }
]

```



```

    }
  ]
}
]

```

EC2CopyImagePolicy

Concede permiso para copiar imágenes de Amazon EC2.

```

"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "ec2:CopyImage"
    ],
    "Resource": {
      "Fn::Sub": [
        "arn:${AWS::Partition}:ec2:${AWS::Region}:${AWS::AccountId}:image/${imageId}",
        {
          "imageId": {
            "Ref": "ImageId"
          }
        }
      ]
    }
  }
]

```

EC2DescribePolicy

Concede permiso para describir instancias de Amazon Elastic Compute Cloud (Amazon EC2).

```

"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "ec2:DescribeRegions",
      "ec2:DescribeInstances"
    ],
    "Resource": "*"
  }
]

```

]

EcsRunTaskPolicy

Concede permiso para iniciar una nueva tarea para una definición de tarea.

```
"Statement": [
  {
    "Action": [
      "ecs:RunTask"
    ],
    "Resource": {
      "Fn::Sub": [
        "arn:${AWS::Partition}:ecs:${AWS::Region}:${AWS::AccountId}:task-definition/
${taskDefinition}",
        {
          "taskDefinition": {
            "Ref": "TaskDefinition"
          }
        }
      ]
    },
    "Effect": "Allow"
  }
]
```

EFSWriteAccessPolicy

Concede permiso para montar un sistema de archivos de Amazon EFS con acceso de escritura.

```
"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "elasticfilesystem:ClientMount",
      "elasticfilesystem:ClientWrite"
    ],
    "Resource": {
      "Fn::Sub": [
        "arn:${AWS::Partition}:elasticfilesystem:${AWS::Region}:${AWS::AccountId}:file-
system/${FileSystem}",
        {
          "FileSystem": {

```

```

        "Ref": "FileSystem"
      }
    }
  ],
  "Condition": {
    "StringEquals": {
      "elasticfilesystem:AccessPointArn": {
        "Fn::Sub": [
          "arn:${AWS::Partition}:elasticfilesystem:${AWS::Region}:
${AWS::AccountId}:access-point/${AccessPoint}",
          {
            "AccessPoint": {
              "Ref": "AccessPoint"
            }
          }
        ]
      }
    }
  }
}
]

```

EKSDescribePolicy

Concede permiso para describir o enumerar clústeres de Amazon Elastic Kubernetes Service (Amazon EKS).

```

"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "eks:DescribeCluster",
      "eks:ListClusters"
    ],
    "Resource": "*"
  }
]

```

ElasticMapReduceAddJobFlowStepsPolicy

Concede permiso para añadir nuevos pasos a un clúster en ejecución.

```
"Statement": [  
  {  
    "Action": "elasticmapreduce:AddJobFlowSteps",  
    "Resource": {  
      "Fn::Sub": [  
        "arn:${AWS::Partition}:elasticmapreduce:${AWS::Region}:  
${AWS::AccountId}:cluster/${clusterId}",  
        {  
          "clusterId": {  
            "Ref": "ClusterId"  
          }  
        }  
      ]  
    },  
    "Effect": "Allow"  
  }  
]
```

ElasticMapReduceCancelStepsPolicy

Concede permiso para cancelar un paso o pasos pendientes en un clúster en ejecución.

```
"Statement": [  
  {  
    "Action": "elasticmapreduce:CancelSteps",  
    "Resource": {  
      "Fn::Sub": [  
        "arn:${AWS::Partition}:elasticmapreduce:${AWS::Region}:  
${AWS::AccountId}:cluster/${clusterId}",  
        {  
          "clusterId": {  
            "Ref": "ClusterId"  
          }  
        }  
      ]  
    },  
    "Effect": "Allow"  
  }  
]
```

ElasticMapReduceModifyInstanceFleetPolicy

Concede permiso para enumerar detalles y modificar las capacidades para las flotas de instancia de un clúster.

```
"Statement": [  
  {  
    "Action": [  
      "elasticmapreduce:ModifyInstanceFleet",  
      "elasticmapreduce:ListInstanceFleets"  
    ],  
    "Resource": {  
      "Fn::Sub": [  
        "arn:${AWS::Partition}:elasticmapreduce:${AWS::Region}:  
${AWS::AccountId}:cluster/${clusterId}",  
        {  
          "clusterId": {  
            "Ref": "ClusterId"  
          }  
        }  
      ]  
    },  
    "Effect": "Allow"  
  }  
]
```

ElasticMapReduceModifyInstanceGroupsPolicy

Concede permiso para enumerar los detalles y modificar la configuración de los grupos de instancias de un clúster.

```
"Statement": [  
  {  
    "Action": [  
      "elasticmapreduce:ModifyInstanceGroups",  
      "elasticmapreduce:ListInstanceGroups"  
    ],  
    "Resource": {  
      "Fn::Sub": [  
        "arn:${AWS::Partition}:elasticmapreduce:${AWS::Region}:  
${AWS::AccountId}:cluster/${clusterId}",  
        {  
          "clusterId": {  

```

```

        "Ref": "ClusterId"
      }
    }
  ],
  "Effect": "Allow"
}
]

```

ElasticMapReduceSetTerminationProtectionPolicy

Concede permiso para establecer la protección de terminación para un clúster.

```

"Statement": [
  {
    "Action": "elasticmapreduce:SetTerminationProtection",
    "Resource": {
      "Fn::Sub": [
        "arn:${AWS::Partition}:elasticmapreduce:${AWS::Region}:
${AWS::AccountId}:cluster/${clusterId}",
        {
          "clusterId": {
            "Ref": "ClusterId"
          }
        }
      ]
    },
    "Effect": "Allow"
  }
]

```

ElasticMapReduceTerminateJobFlowsPolicy

Concede permiso para cerrar un clúster.

```

"Statement": [
  {
    "Action": "elasticmapreduce:TerminateJobFlows",
    "Resource": {
      "Fn::Sub": [
        "arn:${AWS::Partition}:elasticmapreduce:${AWS::Region}:
${AWS::AccountId}:cluster/${clusterId}",

```

```

        {
            "clusterId": {
                "Ref": "ClusterId"
            }
        }
    ],
    "Effect": "Allow"
}
]

```

ElasticsearchHttpPostPolicy

Otorga permisos POST y PUT a Amazon OpenSearch Service.

```

"Statement": [
    {
        "Effect": "Allow",
        "Action": [
            "es:ESHttpPost",
            "es:ESHttpPut"
        ],
        "Resource": {
            "Fn::Sub": [
                "arn:${AWS::Partition}:es:${AWS::Region}:${AWS::AccountId}:domain/${domainName}/*",
                {
                    "domainName": {
                        "Ref": "DomainName"
                    }
                }
            ]
        }
    }
]

```

EventBridgePutEventsPolicy

Otorga permisos para enviar eventos a Amazon EventBridge.

```

"Statement": [
    {
        "Effect": "Allow",

```

```

    "Action": "events:PutEvents",
    "Resource": {
      "Fn::Sub": [
        "arn:${AWS::Partition}:events:${AWS::Region}:${AWS::AccountId}:event-bus/
        ${eventBusName}",
        {
          "eventBusName": {
            "Ref": "EventBusName"
          }
        }
      ]
    }
  }
}
]

```

FilterLogEventsPolicy

Otorga permiso para filtrar CloudWatch los eventos de los registros de un grupo de registros específico.

```

"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "logs:FilterLogEvents"
    ],
    "Resource": {
      "Fn::Sub": [
        "arn:${AWS::Partition}:logs:${AWS::Region}:${AWS::AccountId}:log-group:
        ${logGroupName}:log-stream:*",
        {
          "logGroupName": {
            "Ref": "LogGroupName"
          }
        }
      ]
    }
  }
]

```

FirehoseCrudPolicy

Concede permiso para crear, escribir, actualizar y eliminar una secuencia de entrega de Firehose.


```

"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "firehose:CreateDeliveryStream",
      "firehose>DeleteDeliveryStream",
      "firehose:DescribeDeliveryStream",
      "firehose:PutRecord",
      "firehose:PutRecordBatch",
      "firehose:UpdateDestination"
    ],
    "Resource": {
      "Fn::Sub": [
        "arn:${AWS::Partition}:firehose:${AWS::Region}:",
        "${AWS::AccountId}:deliverystream/${deliveryStreamName}",
        {
          "deliveryStreamName": {
            "Ref": "DeliveryStreamName"
          }
        }
      ]
    }
  }
]

```

FirehoseWritePolicy

Concede permiso para escribir en una secuencia de entrega de Firehose.

```

"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "firehose:PutRecord",
      "firehose:PutRecordBatch"
    ],
    "Resource": {
      "Fn::Sub": [
        "arn:${AWS::Partition}:firehose:${AWS::Region}:",
        "${AWS::AccountId}:deliverystream/${deliveryStreamName}",
        {
          "deliveryStreamName": {
            "Ref": "DeliveryStreamName"
          }
        }
      ]
    }
  }
]

```

```

    }
  }
]
}
]

```

KinesisCrudPolicy

Concede permiso para crear, publicar y eliminar un flujo de Amazon Kinesis.

```

"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "kinesis:AddTagsToStream",
      "kinesis:CreateStream",
      "kinesis:DecreaseStreamRetentionPeriod",
      "kinesis>DeleteStream",
      "kinesis:DescribeStream",
      "kinesis:DescribeStreamSummary",
      "kinesis:GetShardIterator",
      "kinesis:IncreaseStreamRetentionPeriod",
      "kinesis:ListTagsForStream",
      "kinesis:MergeShards",
      "kinesis:PutRecord",
      "kinesis:PutRecords",
      "kinesis:SplitShard",
      "kinesis:RemoveTagsFromStream"
    ],
    "Resource": {
      "Fn::Sub": [
        "arn:${AWS::Partition}:kinesis:${AWS::Region}:${AWS::AccountId}:stream/${streamName}",
        {
          "streamName": {
            "Ref": "StreamName"
          }
        }
      ]
    }
  }
]

```

KinesisStreamReadPolicy

Concede permiso para listar y leer un flujo de Amazon Kinesis.

```
"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "kinesis:ListStreams",
      "kinesis:DescribeLimits"
    ],
    "Resource": {
      "Fn::Sub": "arn:${AWS::Partition}:kinesis:${AWS::Region}:
${AWS::AccountId}:stream/*"
    }
  },
  {
    "Effect": "Allow",
    "Action": [
      "kinesis:DescribeStream",
      "kinesis:DescribeStreamSummary",
      "kinesis:GetRecords",
      "kinesis:GetShardIterator"
    ],
    "Resource": {
      "Fn::Sub": [
        "arn:${AWS::Partition}:kinesis:${AWS::Region}:${AWS::AccountId}:stream/
${streamName}",
        {
          "streamName": {
            "Ref": "StreamName"
          }
        }
      ]
    }
  }
]
```

KMSDecryptPolicy

Da permiso para descifrar con una clave AWS Key Management Service (AWS KMS). Tenga en cuenta que `keyId` debe ser un identificador de AWS KMS clave y no un alias de clave.

```

"Statement": [
  {
    "Action": "kms:Decrypt",
    "Effect": "Allow",
    "Resource": {
      "Fn::Sub": [
        "arn:${AWS::Partition}:kms:${AWS::Region}:${AWS::AccountId}:key/${keyId}",
        {
          "keyId": {
            "Ref": "KeyId"
          }
        }
      ]
    }
  }
]

```

KMSEncryptPolicy

Da permiso para cifrar con una AWS KMS clave. Tenga en cuenta que KeyID debe ser un identificador de AWS KMS clave y no un alias de clave.

```

"Statement": [
  {
    "Action": "kms:Encrypt",
    "Effect": "Allow",
    "Resource": {
      "Fn::Sub": [
        "arn:${AWS::Partition}:kms:${AWS::Region}:${AWS::AccountId}:key/${keyId}",
        {
          "keyId": {
            "Ref": "KeyId"
          }
        }
      ]
    }
  }
]

```

LambdaInvokePolicy

Da permiso para invocar una AWS Lambda función, un alias o una versión.

```

"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "lambda:InvokeFunction"
    ],
    "Resource": {
      "Fn::Sub": [
        "arn:${AWS::Partition}:lambda:${AWS::Region}:${AWS::AccountId}:function:
${functionName}*",
        {
          "functionName": {
            "Ref": "FunctionName"
          }
        }
      ]
    }
  }
]

```

MobileAnalyticsWriteOnlyAccessPolicy

Concede permiso de solo escritura para poner datos de eventos para todos los recursos de la aplicación.

```

"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "mobileanalytics:PutEvents"
    ],
    "Resource": "*"
  }
]

```

OrganizationsListAccountsPolicy

Concede permisos de solo lectura para incluir los nombres e ID de las cuentas secundarias.

```

"Statement": [
  {
    "Effect": "Allow",

```

```
"Action": [
  "organizations:ListAccounts"
],
"Resource": "*"
}
]
```

PinpointEndpointAccessPolicy

Concede permiso para obtener y actualizar puntos de conexión para una aplicación de Amazon Pinpoint.

```
"Statement": [
{
  "Effect": "Allow",
  "Action": [
    "mobiletargeting:GetEndpoint",
    "mobiletargeting:UpdateEndpoint",
    "mobiletargeting:UpdateEndpointsBatch"
  ],
  "Resource": {
    "Fn::Sub": [
      "arn:${AWS::Partition}:mobiletargeting:${AWS::Region}:${AWS::AccountId}:apps/${pinpointApplicationId}/endpoints/*",
      {
        "pinpointApplicationId": {
          "Ref": "PinpointApplicationId"
        }
      }
    ]
  }
}
]
```

PollyFullAccessPolicy

Concede permiso de acceso completo a los recursos del léxico de Amazon Polly.

```
"Statement": [
{
  "Effect": "Allow",
  "Action": [
    "polly:GetLexicon",

```

```

    "polly:DeleteLexicon"
  ],
  "Resource": [
    {
      "Fn::Sub": [
        "arn:${AWS::Partition}:polly:${AWS::Region}:${AWS::AccountId}:lexicon/
${lexiconName}",
        {
          "lexiconName": {
            "Ref": "LexiconName"
          }
        }
      ]
    }
  ]
},
{
  "Effect": "Allow",
  "Action": [
    "polly:DescribeVoices",
    "polly:ListLexicons",
    "polly:PutLexicon",
    "polly:SynthesizeSpeech"
  ],
  "Resource": [
    {
      "Fn::Sub": "arn:${AWS::Partition}:polly:${AWS::Region}:
${AWS::AccountId}:lexicon/*"
    }
  ]
}
]

```

RekognitionDetectOnlyPolicy

Concede permiso para detectar rostros, etiquetas y texto.

```

"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "rekognition:DetectFaces",
      "rekognition:DetectLabels",
      "rekognition:DetectModerationLabels",

```

```
    "rekognition:DetectText"
  ],
  "Resource": "*"
}
```

RekognitionFacesManagementPolicy

Concede permiso para añadir, eliminar y buscar rostros en una colección de Amazon Rekognition.

```
"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "rekognition:IndexFaces",
      "rekognition:DeleteFaces",
      "rekognition:SearchFaces",
      "rekognition:SearchFacesByImage",
      "rekognition:ListFaces"
    ],
    "Resource": {
      "Fn::Sub": [
        "arn:${AWS::Partition}:rekognition:${AWS::Region}:${AWS::AccountId}:collection/${collectionId}",
        {
          "collectionId": {
            "Ref": "CollectionId"
          }
        }
      ]
    }
  }
]
```

RekognitionFacesPolicy

Concede permiso para comparar y detectar rostros y etiquetas.

```
"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "rekognition:CompareFaces",
```



```

    "rekognition:DetectFaces"
  ],
  "Resource": "*"
}
]
```

RekognitionLabelsPolicy

Concede permiso para detectar etiquetas de objetos y de moderación.

```

"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "rekognition:DetectLabels",
      "rekognition:DetectModerationLabels"
    ],
    "Resource": "*"
  }
]
```

RekognitionNoDataAccessPolicy

Concede permiso para comparar y detectar rostros y etiquetas.

```

"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "rekognition:CompareFaces",
      "rekognition:DetectFaces",
      "rekognition:DetectLabels",
      "rekognition:DetectModerationLabels"
    ],
    "Resource": {
      "Fn::Sub": [
        "arn:${AWS::Partition}:rekognition:${AWS::Region}:${AWS::AccountId}:collection/${collectionId}",
        {
          "collectionId": {
            "Ref": "CollectionId"
          }
        }
      ]
    }
  }
]
```

```
    ]
  }
}
]
```

RekognitionReadPolicy

Concede permiso para enumerar y buscar rostros.

```
"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "rekognition:ListCollections",
      "rekognition:ListFaces",
      "rekognition:SearchFaces",
      "rekognition:SearchFacesByImage"
    ],
    "Resource": {
      "Fn::Sub": [
        "arn:${AWS::Partition}:rekognition:${AWS::Region}:${AWS::AccountId}:collection/${collectionId}",
        {
          "collectionId": {
            "Ref": "CollectionId"
          }
        }
      ]
    }
  }
]
```

RekognitionWriteOnlyAccessPolicy

Concede permiso para crear colecciones e indexar rostros.

```
"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "rekognition:CreateCollection",
      "rekognition:IndexFaces"
    ],
  },
]
```

```

    "Resource": {
      "Fn::Sub": [
        "arn:${AWS::Partition}:rekognition:${AWS::Region}:${AWS::AccountId}:collection/${collectionId}",
        {
          "collectionId": {
            "Ref": "CollectionId"
          }
        }
      ]
    }
  }
}
]

```

Route53ChangeResourceRecordSetsPolicy

Concede permiso para cambiar los conjuntos de registros de recursos en Route 53.

```

"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "route53:ChangeResourceRecordSets"
    ],
    "Resource": {
      "Fn::Sub": [
        "arn:${AWS::Partition}:route53::hostedzone/${HostedZoneId}",
        {
          "HostedZoneId": {
            "Ref": "HostedZoneId"
          }
        }
      ]
    }
  }
]

```

S3CrudPolicy

Concede permisos de creación, lectura, actualización y eliminación para actuar en los objetos de un bucket de Amazon S3.

```

"Statement": [

```

```

{
  "Effect": "Allow",
  "Action": [
    "s3:GetObject",
    "s3:ListBucket",
    "s3:GetBucketLocation",
    "s3:GetObjectVersion",
    "s3:PutObject",
    "s3:PutObjectAcl",
    "s3:GetLifecycleConfiguration",
    "s3:PutLifecycleConfiguration",
    "s3:DeleteObject"
  ],
  "Resource": [
    {
      "Fn::Sub": [
        "arn:${AWS::Partition}:s3:::${bucketName}",
        {
          "bucketName": {
            "Ref": "BucketName"
          }
        }
      ]
    },
    {
      "Fn::Sub": [
        "arn:${AWS::Partition}:s3:::${bucketName}/*",
        {
          "bucketName": {
            "Ref": "BucketName"
          }
        }
      ]
    }
  ]
}
]

```

S3FullAccessPolicy

Concede permiso de acceso completo para actuar sobre los objetos de un bucket de Amazon S3.

```

"Statement": [
  {

```

```

    "Effect": "Allow",
    "Action": [
        "s3:GetObject",
        "s3:GetObjectAcl",
        "s3:GetObjectVersion",
        "s3:PutObject",
        "s3:PutObjectAcl",
        "s3:DeleteObject",
        "s3:DeleteObjectTagging",
        "s3:DeleteObjectVersionTagging",
        "s3:GetObjectTagging",
        "s3:GetObjectVersionTagging",
        "s3:PutObjectTagging",
        "s3:PutObjectVersionTagging"
    ],
    "Resource": [
        {
            "Fn::Sub": [
                "arn:${AWS::Partition}:s3:::${bucketName}/*",
                {
                    "bucketName": {
                        "Ref": "BucketName"
                    }
                }
            ]
        }
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "s3:ListBucket",
        "s3:GetBucketLocation",
        "s3:GetLifecycleConfiguration",
        "s3:PutLifecycleConfiguration"
    ],
    "Resource": [
        {
            "Fn::Sub": [
                "arn:${AWS::Partition}:s3:::${bucketName}",
                {
                    "bucketName": {
                        "Ref": "BucketName"
                    }
                }
            ]
        }
    ]
}

```

```

    }
  ]
}
]

```

S3ReadPolicy

Concede permiso de solo lectura para leer objetos en un bucket de Amazon Simple Storage Service (Amazon S3).

```

"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "s3:GetObject",
      "s3:ListBucket",
      "s3:GetBucketLocation",
      "s3:GetObjectVersion",
      "s3:GetLifecycleConfiguration"
    ],
    "Resource": [
      {
        "Fn::Sub": [
          "arn:${AWS::Partition}:s3:::${bucketName}",
          {
            "bucketName": {
              "Ref": "BucketName"
            }
          }
        ]
      },
      {
        "Fn::Sub": [
          "arn:${AWS::Partition}:s3:::${bucketName}/*",
          {
            "bucketName": {
              "Ref": "BucketName"
            }
          }
        ]
      }
    ]
  }
]

```

```

    ]
  }
]

```

S3WritePolicy

Concede permiso de escritura para escribir objetos en un bucket de Amazon S3.

```

"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "s3:PutObject",
      "s3:PutObjectAcl",
      "s3:PutLifecycleConfiguration"
    ],
    "Resource": [
      {
        "Fn::Sub": [
          "arn:${AWS::Partition}:s3:::${bucketName}",
          {
            "bucketName": {
              "Ref": "BucketName"
            }
          }
        ]
      },
      {
        "Fn::Sub": [
          "arn:${AWS::Partition}:s3:::${bucketName}/*",
          {
            "bucketName": {
              "Ref": "BucketName"
            }
          }
        ]
      }
    ]
  }
]

```

SageMakerCreateEndpointConfigPolicy

Otorga permiso para crear una configuración de punto final en SageMaker AI.

```
"Statement": [
  {
    "Action": [
      "sagemaker:CreateEndpointConfig"
    ],
    "Resource": {
      "Fn::Sub": [
        "arn:${AWS::Partition}:sagemaker:${AWS::Region}:${AWS::AccountId}:endpoint-
config/${endpointConfigName}",
        {
          "endpointConfigName": {
            "Ref": "EndpointConfigName"
          }
        }
      ]
    },
    "Effect": "Allow"
  }
]
```

SageMakerCreateEndpointPolicy

Da permiso para crear un punto final en la SageMaker IA.

```
"Statement": [
  {
    "Action": [
      "sagemaker:CreateEndpoint"
    ],
    "Resource": {
      "Fn::Sub": [
        "arn:${AWS::Partition}:sagemaker:${AWS::Region}:${AWS::AccountId}:endpoint/
${endpointName}",
        {
          "endpointName": {
            "Ref": "EndpointName"
          }
        }
      ]
    }
  }
]
```



```
    },
    "Effect": "Allow"
  }
]
```

ServerlessRepoReadWriteAccessPolicy

Otorga permiso para crear y enumerar aplicaciones en el servicio AWS Serverless Application Repository (AWS SAM).

```
"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "serverlessrepo:CreateApplication",
      "serverlessrepo:CreateApplicationVersion",
      "serverlessrepo:GetApplication",
      "serverlessrepo:ListApplications",
      "serverlessrepo:ListApplicationVersions"
    ],
    "Resource": [
      {
        "Fn::Sub": "arn:${AWS::Partition}:serverlessrepo:${AWS::Region}:${AWS::AccountId}:applications/*"
      }
    ]
  }
]
```

SESBulkTemplatedCrudPolicy

Concede permiso para enviar correos electrónicos, correos electrónicos con plantillas y correos electrónicos masivos con plantillas de Amazon SES y para verificar la identidad.

Note

La acción `ses:SendTemplatedEmail` requiere un ARN de plantilla. En su lugar, usa `SESBulkTemplatedCrudPolicy_v2`.

```
"Statement": [
```

```

{
  "Effect": "Allow",
  "Action": [
    "ses:GetIdentityVerificationAttributes",
    "ses:SendEmail",
    "ses:SendRawEmail",
    "ses:SendTemplatedEmail",
    "ses:SendBulkTemplatedEmail",
    "ses:VerifyEmailIdentity"
  ],
  "Resource": {
    "Fn::Sub": [
      "arn:${AWS::Partition}:ses:${AWS::Region}:${AWS::AccountId}:identity/${identityName}",
      {
        "identityName": {
          "Ref": "IdentityName"
        }
      }
    ]
  }
}
]

```

SESBulkTemplatedCrudPolicy_v2

Concede permiso para enviar correos electrónicos, correos electrónicos con plantillas y correos electrónicos masivos con plantillas de Amazon SES y para verificar la identidad.

```

"Statement": [
  {
    "Action": [
      "ses:SendEmail",
      "ses:SendRawEmail",
      "ses:SendTemplatedEmail",
      "ses:SendBulkTemplatedEmail"
    ],
    "Effect": "Allow",
    "Resource": [
      {
        "Fn::Sub": [
          "arn:${AWS::Partition}:ses:${AWS::Region}:${AWS::AccountId}:identity/${identityName}",

```

```

        {
            "identityName": {
                "Ref": "IdentityName"
            }
        }
    ],
},
{
    "Fn::Sub": [
        "arn:${AWS::Partition}:ses:${AWS::Region}:${AWS::AccountId}:template/
${templateName}",
        {
            "templateName": {
                "Ref": "TemplateName"
            }
        }
    ]
}
],
},
{
    "Action": [
        "ses:GetIdentityVerificationAttributes",
        "ses:VerifyEmailIdentity"
    ],
    "Effect": "Allow",
    "Resource": "*"
}
]

```

SESCrudPolicy

Concede permiso para enviar correo electrónico y verificar la identidad.

```

"Statement": [
    {
        "Effect": "Allow",
        "Action": [
            "ses:GetIdentityVerificationAttributes",
            "ses:SendEmail",
            "ses:SendRawEmail",
            "ses:VerifyEmailIdentity"
        ],
        "Resource": {

```

```

    "Fn::Sub": [
      "arn:${AWS::Partition}:ses:${AWS::Region}:${AWS::AccountId}:identity/
${identityName}",
      {
        "identityName": {
          "Ref": "IdentityName"
        }
      }
    ]
  }
}
]

```

SESEmailTemplateCrudPolicy

Concede permiso para crear, obtener, enumerar, actualizar y eliminar plantillas de correo electrónico de Amazon SES.

```

"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "ses:CreateTemplate",
      "ses:GetTemplate",
      "ses:ListTemplates",
      "ses:UpdateTemplate",
      "ses>DeleteTemplate",
      "ses:TestRenderTemplate"
    ],
    "Resource": "*"
  }
]

```

SESSendBouncePolicy

Concede SendBounce permiso a una identidad de Amazon Simple Email Service (Amazon SES).

```

"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "ses:SendBounce"
    ]
  }
]

```

```

    ],
    "Resource": {
      "Fn::Sub": [
        "arn:${AWS::Partition}:ses:${AWS::Region}:${AWS::AccountId}:identity/
        ${identityName}",
        {
          "identityName": {
            "Ref": "IdentityName"
          }
        }
      ]
    }
  }
}
]

```

SNSCrudPolicy

Concede permisos para crear, publicar y suscribirse a temas de Amazon SNS.

```

"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "sns:ListSubscriptionsByTopic",
      "sns:CreateTopic",
      "sns:SetTopicAttributes",
      "sns:Subscribe",
      "sns:Publish"
    ],
    "Resource": {
      "Fn::Sub": [
        "arn:${AWS::Partition}:sns:${AWS::Region}:${AWS::AccountId}:${topicName}*",
        {
          "topicName": {
            "Ref": "TopicName"
          }
        }
      ]
    }
  }
]

```

SNSPublishMessagePolicy

Concede permiso para publicar un mensaje en un tema de Amazon Simple Notification Service (Amazon SNS).

```
"Statement": [  
  {  
    "Effect": "Allow",  
    "Action": [  
      "sns:Publish"  
    ],  
    "Resource": {  
      "Fn::Sub": [  
        "arn:${AWS::Partition}:sns:${AWS::Region}:${AWS::AccountId}:${topicName}",  
        {  
          "topicName": {  
            "Ref": "TopicName"  
          }  
        }  
      ]  
    }  
  }  
]
```

SQSPollerPolicy

Concede permiso para sondear una cola de Amazon Simple Queue Service (Amazon SQS).

```
"Statement": [  
  {  
    "Effect": "Allow",  
    "Action": [  
      "sqs:ChangeMessageVisibility",  
      "sqs:ChangeMessageVisibilityBatch",  
      "sqs:DeleteMessage",  
      "sqs:DeleteMessageBatch",  
      "sqs:GetQueueAttributes",  
      "sqs:ReceiveMessage"  
    ],  
    "Resource": {  
      "Fn::Sub": [  
        "arn:${AWS::Partition}:sqs:${AWS::Region}:${AWS::AccountId}:${queueName}",  
        {  

```

```

        "queueName": {
            "Ref": "QueueName"
        }
    }
]
}
}
]

```

SQSSendMessagePolicy

Concede permisos para enviar mensajes a una cola de Amazon SQS.

```

"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "sqs:SendMessage*"
    ],
    "Resource": {
      "Fn::Sub": [
        "arn:${AWS::Partition}:sqs:${AWS::Region}:${AWS::AccountId}:${queueName}",
        {
          "queueName": {
            "Ref": "QueueName"
          }
        }
      ]
    }
  }
]

```

SSMParameterReadPolicy

Concede permiso para acceder a un parámetro de un almacén de parámetros de Amazon EC2 Systems Manager (SSM) para cargar secretos en esta cuenta. Se utiliza cuando el nombre del parámetro no tiene un prefijo de barra inclinada.

Note

Si no utiliza la clave predeterminada, también necesitará la política `KMSDecryptPolicy`.

```

"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "ssm:DescribeParameters"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "ssm:GetParameters",
      "ssm:GetParameter",
      "ssm:GetParametersByPath"
    ],
    "Resource": {
      "Fn::Sub": [
        "arn:${AWS::Partition}:ssm:${AWS::Region}:${AWS::AccountId}:parameter/
${parameterName}",
        {
          "parameterName": {
            "Ref": "ParameterName"
          }
        }
      ]
    }
  }
]

```

SSMParameterWithSlashPrefixReadPolicy

Concede permiso para acceder a un parámetro de un almacén de parámetros de Amazon EC2 Systems Manager (SSM) para cargar secretos en esta cuenta. Se utiliza cuando el nombre del parámetro tiene un prefijo de barra diagonal.

Note

Si no utiliza la clave predeterminada, también necesitará la política `KMSDecryptPolicy`.

```

"Statement": [
  {

```



```

    "Effect": "Allow",
    "Action": [
        "ssm:DescribeParameters"
    ],
    "Resource": "*"
},
{
    "Effect": "Allow",
    "Action": [
        "ssm:GetParameters",
        "ssm:GetParameter",
        "ssm:GetParametersByPath"
    ],
    "Resource": {
        "Fn::Sub": [
            "arn:${AWS::Partition}:ssm:${AWS::Region}:${AWS::AccountId}:parameter:${parameterName}",
            {
                "parameterName": {
                    "Ref": "ParameterName"
                }
            }
        ]
    }
}
]

```

StepFunctionsExecutionPolicy

Concede permiso para iniciar la ejecución de una máquina de estado de Step Functions.

```

"Statement": [
    {
        "Effect": "Allow",
        "Action": [
            "states:StartExecution"
        ],
        "Resource": {
            "Fn::Sub": [
                "arn:${AWS::Partition}:states:${AWS::Region}:${AWS::AccountId}:stateMachine:${stateMachineName}",
                {
                    "stateMachineName": {
                        "Ref": "StateMachineName"
                    }
                }
            ]
        }
    }
]

```

```
    }  
  }  
]  
}  
}  
]
```

TextractDetectAnalyzePolicy

Concede acceso para detectar y analizar documentos con Amazon Textract.

```
"Statement": [  
  {  
    "Effect": "Allow",  
    "Action": [  
      "textract:DetectDocumentText",  
      "textract:StartDocumentTextDetection",  
      "textract:StartDocumentAnalysis",  
      "textract:AnalyzeDocument"  
    ],  
    "Resource": "*"   
  }  
]
```

TextractGetResultPolicy

Concede acceso a documentos detectados y analizados desde Amazon Textract.

```
"Statement": [  
  {  
    "Effect": "Allow",  
    "Action": [  
      "textract:GetDocumentTextDetection",  
      "textract:GetDocumentAnalysis"  
    ],  
    "Resource": "*"   
  }  
]
```

TextractPolicy

Proporciona acceso completo a Amazon Textract.

```
"Statement": [  
  {  
    "Effect": "Allow",  
    "Action": [  
      "textract:*"  
    ],  
    "Resource": "*"   
  }  
]
```

VPCAccessPolicy

Permite crear, eliminar, describir y eliminar interfaces de red elásticas.

```
"Statement": [  
  {  
    "Effect": "Allow",  
    "Action": [  
      "ec2:CreateNetworkInterface",  
      "ec2:DeleteNetworkInterface",  
      "ec2:DescribeNetworkInterfaces",  
      "ec2:DetachNetworkInterface"  
    ],  
    "Resource": "*"   
  }  
]
```

Gestión de AWS SAM permisos con CloudFormation mecanismos

Para controlar el acceso a AWS los recursos, the AWS Serverless Application Model (AWS SAM) puede usar los mismos mecanismos que. CloudFormation Para obtener más información, consulta [Control de acceso con AWS Identity and Access Management](#) en la Guía del usuario de AWS CloudFormation .

Existen tres opciones principales para conceder a un usuario permiso para administrar aplicaciones sin servidor. Cada opción proporciona a los usuarios diferentes niveles de control de acceso.

- Concesión de permisos de administrador.
- Adjunte las políticas AWS gestionadas necesarias.
- Otorgue permisos específicos AWS Identity and Access Management (IAM).

Según la opción que elija, los usuarios solo pueden administrar las aplicaciones sin servidor que contengan AWS recursos a los que tengan permiso de acceso.

En las siguientes secciones se describe cada proceso de manera detallada.

Concesión de permisos de administrador

Si concedes permisos de administrador a un usuario, este podrá administrar aplicaciones sin servidor que contengan cualquier combinación de AWS recursos. Esta es la opción más sencilla, pero también otorga a los usuarios el conjunto más amplio de permisos, lo que les permite realizar acciones con el mayor impacto.

Para obtener más información acerca de cómo conceder permisos de administrador a un usuario, consulta [Creación del primer grupo de usuarios y administradores de IAM](#) en la Guía del usuario de IAM.

Adjunte las políticas AWS administradas necesarias

Puede conceder a los usuarios un subconjunto de permisos mediante [políticas AWS administradas](#), en lugar de conceder permisos de administrador completos. Si usa esta opción, asegúrese de que el conjunto de políticas AWS administradas cubra todas las acciones y los recursos necesarios para las aplicaciones sin servidor que administran los usuarios.

Por ejemplo, las siguientes políticas AWS administradas son suficientes para [implementar la aplicación Hello World de ejemplo](#):

- AWSCloudFormationFullAccess
- IAMFullAcceder
- AWSLambda_FullAccess
- APIGatewayAdministrador de Amazon
- Amazon S3 FullAccess
- Amazon EC2 ContainerRegistryFullAccess

Para obtener más información acerca de cómo adjuntar políticas a un usuario de IAM, consulta [Cambio de los permisos de un usuario de IAM](#) en la Guía del usuario de IAM.

Conceda permisos específicos de IAM.

Para obtener un nivel de control de acceso más detallado, puede conceder permisos de IAM específicos a los usuarios mediante [instrucciones de política](#). Si usas esta opción, asegúrese de

que las instrucciones de políticas cubran todas las acciones y los recursos necesarios para las aplicaciones sin servidor que administran los usuarios.

La mejor práctica con esta opción es denegar a los usuarios el permiso para crear roles, incluidos los roles de ejecución de Lambda, para que no puedan concederse permisos escalados. Por lo tanto, como administrador, primero debe crear [rol de ejecución de Lambda](#) que se especificará en las aplicaciones sin servidor que administrarán los usuarios. Para obtener información sobre la creación de roles de ejecución de Lambda, consulta [Creación de un rol de ejecución en la consola de IAM](#).

Para la [aplicación Hello World de ejemplo](#), AWSLambdaBasicExecutionRole basta con ejecutar la aplicación. Tras crear un rol de ejecución de Lambda, modifique el archivo de AWS SAM plantilla de la aplicación Hello World de ejemplo para añadir la siguiente propiedad al `AWS::Serverless::Function` recurso:

```
Role: lambda-execution-role-arn
```

Una vez implementada la aplicación Hello World modificada, la siguiente instrucción de política otorga permisos suficientes para que los usuarios implementen, actualicen y eliminen la aplicación:

Note

El ejemplo de declaración de política de esta sección le otorga el permiso suficiente para implementar, actualizar y eliminar la [aplicación Hello World de ejemplo](#). Si agrega tipos de recursos adicionales a su aplicación, debe actualizar la instrucción de política para incluir lo siguiente:

1. Permiso para que su aplicación invoque las acciones del servicio.
2. La entidad principal de servicio, si es necesaria para las acciones del servicio.

Por ejemplo, si agrega un flujo de trabajo de Step Functions, es posible que deba agregar permisos para las acciones que se enumeran [aquí](#) y la entidad principal de servicio `states.amazonaws.com`.

Para obtener más información acerca de las políticas de IAM, consulta [Administración de las políticas de IAM](#) en la Guía del usuario de IAM.

Controle el acceso a la API con su AWS SAM plantilla

Controlar el acceso a las API de API Gateway ayuda a garantizar que su aplicación sin servidor sea segura y solo se pueda acceder a ella mediante la autorización que habilite. Puedes habilitar la autorización en tu AWS SAM plantilla para controlar quién puede acceder a tus API de API Gateway.

AWS SAM admite varios mecanismos para controlar el acceso a las API de API Gateway. El conjunto de mecanismos admitidos varía entre los tipos de recursos `AWS::Serverless::HttpApi` y `AWS::Serverless::Api`.

En la tabla siguiente se resumen los mecanismos que admite cada tipo de recurso.

Mecanismos para controlar el acceso	<code>AWS::Serverless::HttpApi</code>	<code>AWS::Serverless::Api</code>
Autorizadores de Lambda	✓	✓
Permisos de IAM		✓
Grupos de usuarios de Amazon Cognito	✓ *	✓
Claves de API		✓
Políticas de recursos		✓
OAuth 2.0/JWT autorizadores	✓	

* Puede utilizar Amazon Cognito como emisor de un token web JSON (JWT) con el tipo de recurso `AWS::Serverless::HttpApi`.

- **Autorizadores Lambda:** un autorizador Lambda (anteriormente conocido como autorizador personalizado) es una función de eventos de Lambda que se proporciona para controlar el acceso a la API. Cuando se llama a su API, esta función de Lambda se invoca con un contexto de solicitud o un token de autorización que proporciona la aplicación cliente. La función de Lambda responde si la persona que llama está autorizada a realizar la operación solicitada.

Tanto el tipo de recurso `AWS::Serverless::HttpApi` como el tipo de recurso `AWS::Serverless::Api` admiten autorizadores de Lambda.

Para obtener más información acerca de los autorizadores de Lambda para `AWS::Serverless::HttpApi`, consulta [Trabajar con autorizadores AWS Lambda para API HTTP](#) en la Guía para desarrolladores de API Gateway. Para obtener más información sobre los autorizadores de Lambda con `AWS::Serverless::Api`, consulta [Uso de autorizadores Lambda](#) en la Guía para desarrolladores de API Gateway.

Para ver ejemplos de autorizadores de Lambda para cualquier tipo de recurso, consulta [Ejemplos de autorizadores Lambda para AWS SAM](#).

- Permisos de IAM: puede controlar quién puede invocar su API mediante los [permisos de AWS Identity and Access Management \(IAM\)](#). Los usuarios que llamen a su API deben estar autenticados con credenciales de IAM. Las llamadas a su API solo se realizan correctamente si hay una política de IAM asociada al usuario de IAM que represente a la persona que llama a la API, un grupo de IAM que contenga al usuario o un rol de IAM que asuma el usuario.

Solo el tipo de recurso `AWS::Serverless::Api` admite los permisos de IAM.

Para obtener más información, consulta [Control del acceso a las API con permisos de IAM](#) en la Guía para desarrolladores de API Gateway. Para ver un ejemplo, consulta [Ejemplo de permiso de IAM para AWS SAM](#).

- Grupos de usuarios de Amazon Cognito: los grupos de usuarios de Amazon Cognito son directorios de usuarios de Amazon Cognito. Un cliente de su API primero debe iniciar sesión con un usuario en el grupo de usuarios y obtener una identidad o un token de acceso para el usuario. A continuación, el cliente llama a la API con uno de los tokens devueltos. La llamada a la API solo se realiza correctamente si el token requerido es válido.

El tipo de recurso `AWS::Serverless::Api` admite grupos de usuarios de Amazon Cognito. El tipo de recurso `AWS::Serverless::HttpApi` admite el uso de Amazon Cognito como emisor de JWT.

Para obtener más información, consulta [Control del acceso a una API de REST con grupos de usuarios de Amazon Cognito como autorizador](#) en la Guía para desarrolladores de API Gateway. Para ver un ejemplo, consulta [Ejemplo de grupo de usuarios de Amazon Cognito para AWS SAM](#).

- Claves de API: son valores de cadenas alfanuméricas que distribuye a los clientes desarrolladores de aplicaciones para concederles acceso a su API.

Solo el tipo de recurso `AWS::Serverless::Api` admite claves de API.

Para obtener más información sobre claves de API, consulta el tema [Creación y utilización de planes de uso de claves API](#) en la Guía para desarrolladores de API Gateway. Para ver un ejemplo de claves de API, consulta [Ejemplo de clave de API para AWS SAM](#).

- Políticas basadas en recursos: son documentos de política JSON que se adjuntan a una API de API Gateway. Utilice políticas de recursos para controlar si una entidad principal especificada (normalmente un usuario o rol de IAM) puede invocar la API.

Solo el tipo de recurso `AWS::Serverless::Api` admite las políticas de recursos como mecanismo para controlar el acceso a las API de API Gateway.

Para obtener más información sobre las políticas de recursos, consulta [Controlar el acceso a una API con las políticas de recursos de API Gateway](#) en la Guía para desarrolladores de API Gateway. Para ver ejemplos de políticas de recursos, consulta [Ejemplo de política de recursos para AWS SAM](#).

- OAuth 2.0/JWT autorizadores: puede usar JWT como parte de los marcos [OpenID Connect \(OIDC\) y OAuth 2.0](#) para controlar el acceso a sus API. API Gateway valida los JWT que envían los clientes con solicitudes de la API y permite o deniega las solicitudes basándose en la validación del token y, opcionalmente, de los ámbitos del token.

Solo el tipo de recurso es compatible con OAuth 2.0 `AWS::Serverless::HttpApi` JWT autorizadores.

Para obtener más información, consulta [Control del acceso a las API HTTP con autorizadores de JWT](#) en la Guía para desarrolladores de API Gateway. Para ver un ejemplo, consulta [OAuth 2.0/JWT ejemplo de autorizador para AWS SAM](#).

Elegir un mecanismo para controlar el acceso

El mecanismo que elija utilizar para controlar el acceso a las API de API Gateway depende de varios factores. Por ejemplo, si tiene un proyecto totalmente nuevo sin configurar la autorización ni el control de acceso, los grupos de usuarios de Amazon Cognito podrían ser su mejor opción. Esto se debe a que, al configurar grupos de usuarios, también configura automáticamente la autenticación y el control de acceso.

Sin embargo, si tu aplicación ya tiene configurada la autenticación, la mejor opción podría ser utilizar autorizadores de Lambda. Esto se debe a que puede llamar a su servicio de autenticación actual y devolver un documento de política en función de la respuesta. Además, si tu aplicación requiere

una autenticación personalizada o una lógica de control de acceso que los grupos de usuarios no admiten, los autorizadores de Lambda podrían ser su mejor opción.

Cuando haya elegido qué mecanismo utilizar, consulte la sección correspondiente en la sección correspondiente [Ejemplos](#) para saber cómo configurar AWS SAM la aplicación para que utilice ese mecanismo.

Personalizar respuestas de error

Se puede utilizar AWS SAM para personalizar el contenido de algunas respuestas de error de API Gateway. Solo el tipo de recurso `AWS::Serverless::Api` admite respuestas personalizadas de API Gateway.

Para obtener más información, consulta el tema [Respuestas de Gateway en API Gateway](#) en la Guía para desarrolladores de API Gateway. Para ver un ejemplo de respuestas personalizadas, consulta [Ejemplo de respuesta personalizada para AWS SAM](#).

Ejemplos

- [Ejemplos de autorizadores Lambda para AWS SAM](#)
- [Ejemplo de permiso de IAM para AWS SAM](#)
- [Ejemplo de grupo de usuarios de Amazon Cognito para AWS SAM](#)
- [Ejemplo de clave de API para AWS SAM](#)
- [Ejemplo de política de recursos para AWS SAM](#)
- [OAuth 2.0/JWT ejemplo de autorizador para AWS SAM](#)
- [Ejemplo de respuesta personalizada para AWS SAM](#)

Ejemplos de autorizadores Lambda para AWS SAM

El tipo de recurso `AWS::Serverless::Api` admite dos tipos de autorizadores de Lambda: `TOKEN` y `REQUEST`. Solo el tipo de recurso `AWS::Serverless::HttpApi` admite los autorizadores `REQUEST`. A continuación, se incluyen ejemplos de cada tipo.

Ejemplo de autorizador LAMBDA **TOKEN** (`AWS::Serverless::Api`)

Puede controlar el acceso a sus API definiendo un `TOKEN` autorizador Lambda en su plantilla. AWS SAM Para ello, utilice el tipo de datos [ApiAuth](#).

La siguiente es una sección de AWS SAM plantilla de ejemplo para un autorizador LambdaTOKEN:

 Note

En el siguiente ejemplo, el FunctionRole de SAM se genera implícitamente.

```
Resources:
  MyApi:
    Type: AWS::Serverless::Api
    Properties:
      StageName: Prod
      Auth:
        DefaultAuthorizer: MyLambdaTokenAuthorizer
        Authorizers:
          MyLambdaTokenAuthorizer:
            FunctionArn: !GetAtt MyAuthFunction.Arn

  MyFunction:
    Type: AWS::Serverless::Function
    Properties:
      CodeUri: ./src
      Handler: index.handler
      Runtime: nodejs12.x
      Events:
        GetRoot:
          Type: Api
          Properties:
            RestApiId: !Ref MyApi
            Path: /
            Method: get

  MyAuthFunction:
    Type: AWS::Serverless::Function
    Properties:
      CodeUri: ./src
      Handler: authorizer.handler
      Runtime: nodejs12.x
```

Para obtener más información sobre los autorizadores de Lambda, consulta [Uso de autorizadores Lambda de API Gateway](#) en la Guía para desarrolladores de API Gateway.

Ejemplo del autorizador LAMBDA **REQUEST** (AWS::Serverless::Api)

Puede controlar el acceso a sus API definiendo un REQUEST autorizador Lambda en su plantilla. AWS SAM Para ello, utilice el tipo de datos [ApiAuth](#).

La siguiente es una sección de AWS SAM plantilla de ejemplo para un autorizador LambdaREQUEST:

```
Resources:
  MyApi:
    Type: AWS::Serverless::Api
    Properties:
      StageName: Prod
      Auth:
        DefaultAuthorizer: MyLambdaRequestAuthorizer
        Authorizers:
          MyLambdaRequestAuthorizer:
            FunctionPayloadType: REQUEST
            FunctionArn: !GetAtt MyAuthFunction.Arn
            Identity:
              QueryStrings:
                - auth

  MyFunction:
    Type: AWS::Serverless::Function
    Properties:
      CodeUri: ./src
      Handler: index.handler
      Runtime: nodejs12.x
      Events:
        GetRoot:
          Type: Api
          Properties:
            RestApiId: !Ref MyApi
            Path: /
            Method: get

  MyAuthFunction:
    Type: AWS::Serverless::Function
    Properties:
      CodeUri: ./src
      Handler: authorizer.handler
      Runtime: nodejs12.x
```

Para obtener más información sobre los autorizadores de Lambda, consulta [Uso de autorizadores Lambda de API Gateway](#) en la Guía para desarrolladores de API Gateway.

Ejemplo de autorizador Lambda (AWS::Serverless::HttpApi)

Puede controlar el acceso a sus API HTTP definiendo un autorizador Lambda en su plantilla. AWS SAM Para ello, utilice el tipo de datos [HttpApiAuth](#).

La siguiente es una sección de AWS SAM plantilla de ejemplo para un autorizador Lambda:

```
Resources:
  MyApi:
    Type: AWS::Serverless::HttpApi
    Properties:
      StageName: Prod
      Auth:
        DefaultAuthorizer: MyLambdaRequestAuthorizer
        Authorizers:
          MyLambdaRequestAuthorizer:
            FunctionArn: !GetAtt MyAuthFunction.Arn
            FunctionInvokeRole: !GetAtt MyAuthFunctionRole.Arn
            Identity:
              Headers:
                - Authorization
            AuthorizerPayloadFormatVersion: 2.0
            EnableSimpleResponses: true

  MyFunction:
    Type: AWS::Serverless::Function
    Properties:
      CodeUri: ./src
      Handler: index.handler
      Runtime: nodejs12.x
      Events:
        GetRoot:
          Type: HttpApi
          Properties:
            ApiId: !Ref MyApi
            Path: /
            Method: get
            PayloadFormatVersion: "2.0"

  MyAuthFunction:
```

```
Type: AWS::Serverless::Function
Properties:
  CodeUri: ./src
  Handler: authorizer.handler
  Runtime: nodejs12.x
```

Ejemplo de permiso de IAM para AWS SAM

Puede controlar el acceso a sus API definiendo permisos IAM en su plantilla de AWS SAM . Para ello, utilice el tipo de datos [ApiAuth](#).

A continuación se muestra un ejemplo de AWS SAM plantilla que se utiliza para los permisos de IAM:

```
AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31
Resources:
  MyApi:
    Type: AWS::Serverless::Api
    Properties:
      StageName: Prod
      Description: 'API with IAM authorization'
      Auth:
        DefaultAuthorizer: AWS_IAM #sets AWS_IAM auth for all methods in this API
  MyFunction:
    Type: AWS::Serverless::Function
    Properties:
      Handler: index.handler
      Runtime: python3.10
      Events:
        GetRoot:
          Type: Api
          Properties:
            RestApiId: !Ref MyApi
            Path: /
            Method: get
      InlineCode: |
        def handler(event, context):
          return {'body': 'Hello World!', 'statusCode': 200}
```

Para obtener más información sobre los permisos de IAM, consulta [Controlar el acceso para invocar una API](#) en la Guía para desarrolladores de API Gateway.

Ejemplo de grupo de usuarios de Amazon Cognito para AWS SAM

Puede controlar el acceso a sus API definiendo un grupo de usuarios de Amazon Cognito en su plantilla de AWS SAM . Para ello, utilice el tipo de datos [ApiAuth](#).

A continuación, se muestra un ejemplo AWS SAM de sección de plantillas para un grupo de usuarios:

```
Resources:
  MyApi:
    Type: AWS::Serverless::Api
    Properties:
      StageName: Prod
      Cors: "'*'"
      Auth:
        DefaultAuthorizer: MyCognitoAuthorizer
        Authorizers:
          MyCognitoAuthorizer:
            UserPoolArn: !GetAtt MyCognitoUserPool.Arn

  MyFunction:
    Type: AWS::Serverless::Function
    Properties:
      CodeUri: ./src
      Handler: lambda.handler
      Runtime: nodejs12.x
      Events:
        Root:
          Type: Api
          Properties:
            RestApiId: !Ref MyApi
            Path: /
            Method: GET

  MyCognitoUserPool:
    Type: AWS::Cognito::UserPool
    Properties:
      UserPoolName: !Ref CognitoUserPoolName
      Policies:
        PasswordPolicy:
          MinimumLength: 8
      UsernameAttributes:
        - email
```

```
Schema:
  - AttributeDataType: String
    Name: email
    Required: false

MyCognitoUserPoolClient:
  Type: AWS::Cognito::UserPoolClient
  Properties:
    UserPoolId: !Ref MyCognitoUserPool
    ClientName: !Ref CognitoUserPoolClientName
    GenerateSecret: false
```

Para obtener más información, consulta [Control del acceso a una API de REST con grupos de usuarios de Amazon Cognito como autorizador](#) en la Guía para desarrolladores de API Gateway.

Ejemplo de clave de API para AWS SAM

Puede controlar el acceso a sus API solicitando claves de API en su AWS SAM plantilla. Para ello, utilice el tipo de datos [ApiAuth](#).

A continuación, se muestra un ejemplo de una sección AWS SAM de plantillas para las claves de API:

```
Resources:
  MyApi:
    Type: AWS::Serverless::Api
    Properties:
      StageName: Prod
      Auth:
        ApiKeyRequired: true # sets for all methods

  MyFunction:
    Type: AWS::Serverless::Function
    Properties:
      CodeUri: .
      Handler: index.handler
      Runtime: nodejs12.x
      Events:
        ApiKey:
          Type: Api
          Properties:
            RestApiId: !Ref MyApi
            Path: /
```

```
Method: get
Auth:
  ApiKeyRequired: true
```

Para obtener más información sobre claves de API, consulta el tema [Creación y utilización de planes de uso de claves API](#) en la Guía para desarrolladores de API Gateway.

Ejemplo de política de recursos para AWS SAM

Puede controlar el acceso a sus API adjuntando una política de recursos a su plantilla de AWS SAM . Para ello, utilice el tipo de datos [ApiAuth](#).

El siguiente es un ejemplo AWS SAM de plantilla para una API privada. Una API privada debe tener una política de recursos para poder implementarse.

```
AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31
Resources:
  MyPrivateApi:
    Type: AWS::Serverless::Api
    Properties:
      StageName: Prod
      EndpointConfiguration: PRIVATE # Creates a private API. Resource policies are
      required for all private APIs.
      Auth:
        ResourcePolicy:
          CustomStatements:
            - Effect: 'Allow'
              Action: 'execute-api:Invoke'
              Resource: ['execute-api:/*/*/*']
              Principal: '*'
            - Effect: 'Deny'
              Action: 'execute-api:Invoke'
              Resource: ['execute-api:/*/*/*']
              Principal: '*'
  MyFunction:
    Type: 'AWS::Serverless::Function'
    Properties:
      InlineCode: |
        def handler(event, context):
            return {'body': 'Hello World!', 'statusCode': 200}
      Handler: index.handler
      Runtime: python3.10
```



```

Events:
  AddItem:
    Type: Api
    Properties:
      RestApiId:
        Ref: MyPrivateApi
      Path: /
      Method: get

```

Para obtener más información sobre las políticas de recursos, consulta [Controlar el acceso a una API con las políticas de recursos de API Gateway](#) en la Guía para desarrolladores de API Gateway. Para obtener más información sobre las API privadas, consulte [Creación de una API privada en Amazon API Gateway](#) en la Guía para desarrolladores de API Gateway.

OAuth 2.0/JWT ejemplo de autorizador para AWS SAM

Puede utilizar los JWT como parte de los marcos [OpenID Connect \(OIDC\)](#) y [OAuth 2.0](#) para controlar el acceso a las API. Para ello, utilice el tipo de datos [HttpApiAuth](#).

A continuación se muestra un ejemplo de sección AWS SAM de plantillas para un OAuth 2.0/JWT autorizador:

```

Resources:
  MyApi:
    Type: AWS::Serverless::HttpApi
    Properties:
      Auth:
        Authorizers:
          MyOAuth2Authorizer:
            AuthorizationScopes:
              - scope
            IdentitySource: $request.header.Authorization
            JwtConfiguration:
              audience:
                - audience1
                - audience2
              issuer: "https://www.example.com/v1/connect/oidc"
            DefaultAuthorizer: MyOAuth2Authorizer
        StageName: Prod
  MyFunction:
    Type: AWS::Serverless::Function
    Properties:

```

```

CodeUri: ./src
Events:
  GetRoot:
    Properties:
      ApiId: MyApi
      Method: get
      Path: /
      PayloadFormatVersion: "2.0"
    Type: HttpApi
Handler: index.handler
Runtime: nodejs12.x

```

Para obtener más información sobre OAuth 2.0/JWT autorizadores, consulte [Control del acceso a las API HTTP con autorizadores JWT](#) en la Guía para desarrolladores de API Gateway.

Ejemplo de respuesta personalizada para AWS SAM

Puedes personalizar algunas respuestas de error de API Gateway definiendo los encabezados de respuesta en su plantilla de AWS SAM. Para ello, utilice el tipo de datos [Gateway Response Object](#).

A continuación se muestra un ejemplo AWS SAM de plantilla que crea una respuesta personalizada para el DEFAULT_5XX error.

```

AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31
Resources:
  MyApi:
    Type: AWS::Serverless::Api
    Properties:
      StageName: Prod
      GatewayResponses:
        DEFAULT_5XX:
          ResponseParameters:
            Headers:
              Access-Control-Expose-Headers: "'WWW-Authenticate'"
              Access-Control-Allow-Origin: "'*'"
              ErrorHeader: "'MyCustomErrorHandler'"
          ResponseTemplates:
            application/json: "{\"message\": \"Error on the $context.resourcePath resource\" }"

      GetFunction:
        Type: AWS::Serverless::Function

```

```
Properties:
  Runtime: python3.10
  Handler: index.handler
  InlineCode: |
    def handler(event, context):
      raise Exception('Check out the new response!')
Events:
  GetResource:
    Type: Api
    Properties:
      Path: /error
      Method: get
      RestApiId: !Ref MyApi
```

Para obtener más información, consulta el tema [Respuestas de Gateway en API Gateway](#) en la Guía para desarrolladores de API Gateway.

Aumente la eficiencia con las capas Lambda con AWS SAM

Con AWS SAMél, puede incluir capas en sus aplicaciones sin servidor. AWS Lambda las capas permiten extraer código de una función de eventos de Lambda a una capa de Lambda, que luego se puede utilizar en varias funciones de Lambda. Esto le permite reducir el tamaño de los paquetes de implementación, separar la lógica de las funciones principales de las dependencias y compartir las dependencias entre varias funciones. Para obtener más información acerca de las capas, consulte [Capas Lambda](#), en la Guía para desarrolladores de AWS Lambda .

Este tema proporciona información sobre lo siguiente:

- Incluir capas en la aplicación
- Cómo se almacenan las capas en caché local

Para obtener información sobre la creación de capas personalizadas, consulta [Creación de capas Lambda en AWS SAM](#).

Incluir capas en la aplicación

Para incluir capas en la aplicación, utiliza la propiedad `Layers` del tipo de recurso [AWS::Serverless::Función](#).

A continuación se presenta un ejemplo AWS SAM de plantilla con una función Lambda que incluye una capa:

```
ServerlessFunction:
  Type: AWS::Serverless::Function
  Properties:
    CodeUri: .
    Handler: my_handler
    Runtime: Python3.7
    Layers:
      - <LayerVersion ARN>
```

Cómo se almacenan las capas en caché local

Al invocar la función mediante uno de los comandos `sam local`, el paquete de capas de la función se descarga y se almacena en caché en el host local.

En la tabla siguiente se muestran las ubicaciones predeterminadas de los directorios de caché para los distintos sistemas operativos.

SO	Ubicación
Windows 7	C:\Users\<user>\AppData\Roaming\AWS SAM
Windows 8	C:\Users\<user>\AppData\Roaming\AWS SAM
Windows 10	C:\Users\<user>\AppData\Roaming\AWS SAM
macOS	~/ .aws-sam/layers-pkg
Unix	~/ .aws-sam/layers-pkg

Una vez almacenado el paquete en caché, la CLI de AWS SAM superpone las capas en una imagen de Docker que se utiliza para invocar la función. AWS SAM CLI genera los nombres de las imágenes que crea, así como las LayerVersions que se guardan en la memoria caché. Encontrará más detalles sobre el esquema en las secciones siguientes.

Para inspeccionar las capas superpuestas, ejecuta el siguiente comando para iniciar una sesión de bash en la imagen que desea inspeccionar:

```
docker run -it --entrypoint=/bin/bash samcli/lambda:<Tag following the schema outlined
in Docker Image Tag Schema> -i
```

Esquema de nombres del directorio de almacenamiento en caché de capas

Dado `LayerVersionArn` que a está definido en su plantilla, AWS SAMCLI extrae la versión `LayerName` y del ARN. Crea un directorio para colocar el contenido de la capa denominada `LayerName-Version-<first 10 characters of sha256 of ARN>`.

Ejemplo:

```
ARN = arn:aws:lambda:us-west-2:111111111111:layer:myLayer:1
Directory name = myLayer-1-926eeb5ff1
```

Esquema de etiquetas de Docker Images

Para calcular el hash de las capas únicas, combina todos los nombres de capa únicos con un delimitador de «-», tome el hash SHA256 y, a continuación, tome los primeros 10 caracteres.

Ejemplo:

```
ServerlessFunction:
  Type: AWS::Serverless::Function
  Properties:
    CodeUri: .
    Handler: my_handler
    Runtime: Python3.7
    Layers:
      - arn:aws:lambda:us-west-2:111111111111:layer:myLayer:1
      - arn:aws:lambda:us-west-2:111111111111:layer:mySecondLayer:1
```

Los nombres únicos se calculan de la misma manera que en el esquema de nombres del directorio de almacenamiento en caché de capas:

```
arn:aws:lambda:us-west-2:111111111111:layer:myLayer:1 = myLayer-1-926eeb5ff1
arn:aws:lambda:us-west-2:111111111111:layer:mySecondLayer:1 =
mySecondLayer-1-6bc1022bdf
```

Para calcular el hash de las capas únicas, combina todos los nombres de capa únicos con un delimitador de «-», tome el hash sha256 y, a continuación, tome los primeros 25 caracteres:

```
myLayer-1-926eeb5ff1-mySecondLayer-1-6bc1022bdf = 2dd7ac5ffb30d515926aef
```

A continuación, combina este valor con el tiempo de ejecución y la arquitectura de la función, con un delimitador de '-':

```
python3.7-x86_64-2dd7ac5ffb30d515926aefffd
```

Reutilice el código y los recursos mediante aplicaciones anidadas en AWS SAM

Una aplicación sin servidor puede incluir una o más aplicaciones anidadas. Una aplicación anidada forma parte de una aplicación más grande y se puede empaquetar e implementar como un artefacto independiente o como un componente de una aplicación más grande. Las aplicaciones anidadas permiten convertir el código de uso frecuente en su propia aplicación, que luego se puede reutilizar en una aplicación sin servidor más grande o en varias aplicaciones sin servidor.

A medida que crecen las arquitecturas sin servidor, suelen surgir patrones comunes en los que los mismos componentes se definen en varias plantillas de aplicaciones. Las aplicaciones anidadas le permiten reutilizar el código, la funcionalidad, los recursos y las configuraciones comunes en AWS SAM plantillas independientes, lo que le permite mantener el código de una sola fuente. Esto reduce la duplicación de códigos y configuraciones. Además, este enfoque modular agiliza el desarrollo, mejora la organización del código y facilita la coherencia entre las aplicaciones sin servidor. Con las aplicaciones anidadas, puede concentrarse más en la lógica empresarial exclusiva de tu aplicación.

Para definir una aplicación anidada en tu aplicación sin servidor, utilice el tipo de recurso [AWS::Serverless::Application](#).

Puede definir aplicaciones anidadas a partir de las dos fuentes siguientes:

- Una aplicación de AWS Serverless Application Repository : usted puede definir aplicaciones anidadas mediante las aplicaciones que están disponibles para su cuenta en el AWS Serverless Application Repository. Pueden ser aplicaciones privadas de su cuenta, aplicaciones que se comparten de forma privada con su cuenta o aplicaciones que se comparten públicamente en el AWS Serverless Application Repository. Para obtener más información sobre los distintos niveles de permisos de implementación, consulta los [Permisos de implementación de aplicaciones](#) y la [Publicación de aplicaciones](#) en la Guía para desarrolladores AWS Serverless Application Repository .
- Una aplicación local: usted puede definir aplicaciones anidadas mediante aplicaciones almacenadas en el sistema de archivos local.

Consulte las siguientes secciones para obtener información detallada sobre cómo AWS SAM definir estos dos tipos de aplicaciones anidadas en su aplicación sin servidor.

 Note

El número máximo de aplicaciones que se pueden anidar en una aplicación sin servidor es de 200.

El número máximo de parámetros que puede tener una aplicación anidada es 60.

Definir una aplicación anidada a partir del AWS Serverless Application Repository

Usted puede definir aplicaciones anidadas mediante las aplicaciones que están disponibles en el AWS Serverless Application Repository. También puede almacenar y distribuir aplicaciones que contengan aplicaciones anidadas mediante el AWS Serverless Application Repository. Para revisar los detalles de una aplicación anidada en AWS Serverless Application Repository, puede usar el AWS SDK, la consola Lambda o la AWS CLI consola Lambda.

Para definir una aplicación que esté alojada AWS Serverless Application Repository en la AWS SAM plantilla de su aplicación sin servidor, utilice el botón Copiar como recurso SAM de la página de detalles de cada aplicación. AWS Serverless Application Repository Para ello, sigue estos pasos:

1. Asegúrese de que ha iniciado sesión en Consola de administración de AWS.
2. Busque la aplicación en la que desee insertarla AWS Serverless Application Repository siguiendo los pasos de la sección [Exploración, búsqueda e implementación de aplicaciones](#) de la Guía para AWS Serverless Application Repository desarrolladores.
3. Seleccione el botón Copiar como recurso SAM. La sección de plantillas SAM de la aplicación que está viendo ahora está en su portapapeles.
4. Pegue la sección de la plantilla SAM en la sección de Resources : del archivo de plantilla SAM de la aplicación que desee incluir en esta aplicación.

A continuación se muestra un ejemplo de una sección de plantillas SAM para una aplicación anidada que está alojada en el AWS Serverless Application Repository:

```
Transform: AWS::Serverless-2016-10-31
```

```
Resources:
```

```

applicationaliasname:
  Type: AWS::Serverless::Application
  Properties:
    Location:
      ApplicationId: arn:aws:serverlessrepo:us-
east-1:123456789012:applications/application-alias-name
      SemanticVersion: 1.0.0
    Parameters:
      # Optional parameter that can have default value overridden
      # ParameterName1: 15 # Uncomment to override default value
      # Required parameter that needs value to be provided
      ParameterName2: YOUR_VALUE

```

Si no hay ninguna configuración de parámetros obligatoria, puede omitir la sección `Parameters:` de la plantilla.

Important

Las aplicaciones que contienen aplicaciones anidadas alojadas en ellas AWS Serverless Application Repository heredan las restricciones de uso compartido de las aplicaciones anidadas.

Por ejemplo, supongamos que una aplicación se comparte públicamente, pero contiene una aplicación anidada que solo se comparte de forma privada con la AWS cuenta que creó la aplicación principal. En este caso, si tu AWS cuenta no tiene permiso para implementar la aplicación anidada, no podrás implementar la aplicación principal. Para más información sobre los permisos para implementar aplicaciones, consulta [Permisos de implementación de aplicaciones](#) y [Publicación de aplicaciones](#) en la Guía del desarrollador de AWS Serverless Application Repository .

Definir una aplicación anidada desde el sistema de archivos local

Usted puede definir aplicaciones anidadas mediante aplicaciones almacenadas en el sistema de archivos local. Para ello, especifique la ruta al archivo de AWS SAM plantilla que está almacenado en el sistema de archivos local.

A continuación se muestra un ejemplo de una sección de plantillas SAM para una aplicación local anidada:

```

Transform: AWS::Serverless-2016-10-31

```



```
Resources:
  applicationaliasname:
    Type: AWS::Serverless::Application
    Properties:
      Location: ../my-other-app/template.yaml
      Parameters:
        # Optional parameter that can have default value overridden
        # ParameterName1: 15 # Uncomment to override default value
        # Required parameter that needs value to be provided
        ParameterName2: YOUR_VALUE
```

Si no hay ninguna configuración de parámetros, puede omitir la sección `Parameters:` de la plantilla.

Implementar aplicaciones anidadas

Puede implementar tu aplicación anidada mediante el comando `aws sam deploy` de la AWS SAM CLI. Para obtener más información, consulta [Implemente su aplicación y sus recursos con AWS SAM](#).

Note

Cuando implementa una aplicación que contiene aplicaciones anidadas, debe reconocer que contiene aplicaciones anidadas. Para ello, pase `CAPABILITY_AUTO_EXPAND` a la [CreateCloudFormationChangeSet API](#) o utilice el `aws serverlessrepo create-cloud-formation-change-set` AWS CLI comando.

Para obtener más información sobre cómo reconocer las aplicaciones anidadas, consulta [Reconocimiento de los Roles de IAM, las políticas de recursos y las aplicaciones anidadas al implementar aplicaciones](#) en la Guía para desarrolladores de AWS Serverless Application Repository .

Gestione eventos basados en el tiempo con el EventBridge programador en AWS SAM

El contenido de este tema proporciona detalles sobre qué es Amazon EventBridge Scheduler, qué soporte AWS SAM ofrece, cómo puede crear eventos de Scheduler y ejemplos a los que puede hacer referencia al crear eventos de Scheduler.

¿Qué es Amazon EventBridge Scheduler?

Usa EventBridge Scheduler para programar eventos en tus plantillas. AWS SAM Amazon EventBridge Scheduler es un servicio de programación que le permite crear, iniciar y gestionar decenas de millones de eventos y tareas en todos los AWS servicios. Este servicio es especialmente útil para los eventos relacionados con el tiempo. Puede usarlo para programar eventos e invocaciones periódicas basadas en el tiempo. También admite eventos únicos, así como expresiones cronológicas y de frecuencia con una hora de inicio y finalización.

Para obtener más información sobre Amazon EventBridge Scheduler, consulte [¿Qué es Amazon EventBridge Scheduler?](#) en la Guía del usuario de EventBridge Scheduler.

Temas

- [EventBridge Soporte para el programador en AWS SAM](#)
- [Crear eventos del EventBridge Scheduler en AWS SAM](#)
- [Ejemplos](#)
- [Más información](#)

EventBridge Soporte para el programador en AWS SAM

La especificación de la plantilla AWS Serverless Application Model (AWS SAM) proporciona una sintaxis simple y abreviada que puedes usar para programar eventos con EventBridge Scheduler for y. AWS Lambda AWS Step Functions

Crear eventos del EventBridge Scheduler en AWS SAM

Defina la `ScheduleV2` propiedad como el tipo de evento en su AWS SAM plantilla para definir su evento del EventBridge Scheduler. Esta propiedad admite los tipos de recursos `AWS::Serverless::Function` y `AWS::Serverless::StateMachine`.

```
MyFunction:
  Type: AWS::Serverless::Function
  Properties:
    Events:
      CWSchedule:
        Type: ScheduleV2
        Properties:
          ScheduleExpression: 'rate(1 minute)'
          Name: TestScheduleV2Function
```

Description: Test schedule event

MyStateMachine:

Type: AWS::Serverless::StateMachine

Properties:

Events:

CWSchedule:

Type: ScheduleV2

Properties:

ScheduleExpression: 'rate(1 minute)'

Name: TestScheduleV2StateMachine

Description: Test schedule event

EventBridge La programación de eventos del programador también admite colas con letra muerta (DLQ) para los eventos no procesados. Para obtener más información sobre las colas de letra muerta, consulte [Configuración de una cola de letra muerta para Scheduler en la Guía del usuario de Scheduler](#). EventBridge EventBridge

Cuando se especifica un ARN de DLQ AWS SAM , configura los permisos para que el programador envíe mensajes al DLQ. Cuando no se especifica un ARN de DLQ AWS SAM , se creará el recurso de DLQ.

Ejemplos

Ejemplo básico de definición de un evento de Scheduler con EventBridge AWS SAM

```
Transform: AWS::Serverless-2016-10-31
Resources:
  MyLambdaFunction:
    Type: AWS::Serverless::Function
    Properties:
      Handler: index.handler
      Runtime: python3.8
      InlineCode: |
        def handler(event, context):
            print(event)
            return {'body': 'Hello World!', 'statusCode': 200}
      MemorySize: 128
    Events:
      Schedule:
        Type: ScheduleV2
        Properties:
```

```

    ScheduleExpression: rate(1 minute)
    Input: '{"hello": "simple"}'

MySFNFunction:
  Type: AWS::Serverless::Function
  Properties:
    Handler: index.handler
    Runtime: python3.8
    InlineCode: |
      def handler(event, context):
        print(event)
        return {'body': 'Hello World!', 'statusCode': 200}
    MemorySize: 128

StateMachine:
  Type: AWS::Serverless::StateMachine
  Properties:
    Type: STANDARD
    Definition:
      StartAt: MyLambdaState
      States:
        MyLambdaState:
          Type: Task
          Resource: !GetAtt MySFNFunction.Arn
          End: true
    Policies:
      - LambdaInvokePolicy:
          FunctionName: !Ref MySFNFunction
  Events:
    Schedule:
      Type: ScheduleV2
      Properties:
        ScheduleExpression: rate(1 minute)
        Input: '{"hello": "simple"}'

```

Más información

Para obtener más información sobre la definición de la propiedad ScheduleV2 EventBridge Scheduler, consulte:

- [ScheduleV2](#) para AWS::Serverless::Function.
- [ScheduleV2](#) para AWS::Serverless::StateMachine.

Orquestando AWS SAM recursos con AWS Step Functions

Puede utilizarlos [AWS Step Functions](#) para organizar funciones de AWS Lambda eventos y otros AWS recursos para crear flujos de trabajo complejos y sólidos. Step Functions le indica a la aplicación cuándo y en qué condiciones se utilizan sus AWS recursos, como AWS Lambda las funciones. Esto simplifica el proceso de establecer una conexión entre estos recursos. Con [AWS::Serverless::StateMachine](#), puede definir los pasos individuales de su flujo de trabajo, asociar los recursos en cada paso y, a continuación, secuenciar estos pasos juntos. También puede agregar transiciones y condiciones donde sea necesario. Esto simplifica el proceso de establecer una conexión entre estos recursos.

Note

Para gestionar AWS SAM las plantillas que contienen máquinas de estado de Step Functions, debe utilizar la versión 0.52.0 o posterior de. AWS SAMCLI Puedes comprobar qué versión tienes ejecutando el comando `sam --version`.

Step Functions se basa en los conceptos de [tareas](#) y [máquinas de estado](#). Las máquinas de estado se definen mediante el [lenguaje de los estados de JSON-based Amazon](#). La [consola de Step Functions](#) muestra una vista gráfica de la estructura de la máquina de estado, para que pueda comprobar visualmente la lógica de la máquina de estado y monitorizar las ejecuciones.

Con la compatibilidad con Step Functions en AWS Serverless Application Model (AWS SAM), puede hacer lo siguiente:

- Defina las máquinas de estados, ya sea directamente dentro AWS SAM de una plantilla o en un archivo independiente
- Cree funciones de ejecución de máquinas de estados mediante AWS SAM plantillas de políticas, políticas integradas o políticas gestionadas
- Activa ejecuciones de máquinas de estado con EventBridge eventos de API Gateway o Amazon, según una programación dentro de una AWS SAM plantilla o llamando directamente a las API
- Utilice las [plantillas de políticas de AWS SAM](#) disponibles para los patrones de desarrollo comunes de Step Functions.

Ejemplo

El siguiente fragmento de ejemplo de un archivo de AWS SAM plantilla define una máquina de estados de Step Functions en un archivo de definición. Ten en cuenta que el archivo `my_state_machine.asl.json` debe estar inscrito en [Amazon States Language](#).

```
AWSTemplateFormatVersion: "2010-09-09"
Transform: AWS::Serverless-2016-10-31
Description: Sample SAM template with Step Functions State Machine

Resources:
  MyStateMachine:
    Type: AWS::Serverless::StateMachine
    Properties:
      DefinitionUri: statemachine/my_state_machine.asl.json
      ...
```

Para descargar una AWS SAM aplicación de ejemplo que incluye una máquina de estados de Step Functions, consulte [Creación de una máquina de estados de Step Functions mediante AWS SAM](#) la Guía para AWS Step Functions desarrolladores.

Más información

Para obtener más información sobre Step Functions y su uso con AWS SAM, consulte lo siguiente:

- [Cómo funciona AWS Step Functions](#)
- [AWS Step Functions y AWS Serverless Application Model](#)
- [Tutorial: Crear una máquina de estados de Step Functions usando AWS SAM](#)
- [AWS SAM Especificación: AWS::Serverless:: StateMachine](#)

Configura la firma de código para tu AWS SAM aplicación

Para garantizar que solo se implemente código confiable, puede utilizar esta opción AWS SAM para habilitar la firma de código en sus aplicaciones sin servidor. Firmar el código ayuda a garantizar que el código no se haya modificado desde la firma y que solo los paquetes de código firmados de editores de confianza se ejecuten en sus funciones de eventos de Lambda. Esto ayuda a las organizaciones a liberarse de la carga de crear componentes de control en sus procesos de implementación.

Para obtener más información sobre la característica de firma de código, consulta [Configuración de firma de código para funciones de Lambda](#) en la Guía para desarrolladores de AWS Lambda .

Antes de poder configurar la firma de código para su aplicación sin servidor, debe crear un perfil de firma mediante AWS Signer. Este perfil de firma se utiliza para las siguientes tareas:

1. Creación de una configuración de firma de código: declare un recurso [AWS::Lambda::CodeSigningConfig](#) para especificar los perfiles de firma de los editores de confianza y establecer la acción política para las comprobaciones de validación. Puede declarar este objeto en la misma AWS SAM plantilla que la función sin servidor, en una AWS SAM plantilla diferente o en una plantilla CloudFormation. A continuación, habilita la firma de código para una función sin servidor especificando la propiedad de la función [CodeSigningConfigArn](#) con el nombre de recurso de Amazon (ARN) de un recurso [AWS::Lambda::CodeSigningConfig](#).
2. Firmar el código: utiliza el comando [sam package](#) o [sam deploy](#) con la opción `--signing-profiles`.

 Note

Para poder firmar correctamente el código con los comandos `sam package` o `sam deploy`, el control de versiones debe estar habilitado para el bucket de Amazon S3 que utiliza con estos comandos. Si utiliza el bucket de Amazon S3 que se AWS SAM crea para usted, el control de versiones se habilita automáticamente. Para obtener más información sobre el control de versiones de los buckets de Amazon S3 e instrucciones para habilitar el control de versiones en un bucket de Amazon S3 que proporciones, consulta [Uso del control de versiones en los buckets de Amazon S3](#) en la Guía del usuario de Amazon Simple Storage Service.

Al implementar una aplicación sin servidor, Lambda realiza comprobaciones de validación en todas las funciones para las que ha habilitado la firma de código. Lambda también realiza comprobaciones de validación en cualquier capa de la que dependan esas funciones. Para obtener más información sobre las comprobaciones de validación de Lambda, consulta la [validación de firmas](#) en la Guía para desarrolladores de AWS Lambda .

Ejemplo

Creación de un perfil de firma

Para crearlo, ejecuta este comando:

```
aws signer put-signing-profile --platform-id "AWSLambda-SHA384-ECDSA" --profile-name MySigningProfile
```

Si el comando anterior se ejecuta correctamente, verás que se devuelve el ARN del perfil de firma. Por ejemplo:

```
{
  "arn": "arn:aws:signer:us-east-1:111122223333:/signing-profiles/MySigningProfile",
  "profileVersion": "SAMPLEverx",
  "profileVersionArn": "arn:aws:signer:us-east-1:111122223333:/signing-profiles/MySigningProfile/SAMPLEverx"
}
```

El campo `profileVersionArn` contiene el ARN que se utilizará al crear la configuración de firma de código.

Creación de una configuración de firma de código para una función

La siguiente AWS SAM plantilla de ejemplo declara un

[AWS::Lambda::CodeSigningConfig](#) recurso y habilita la firma de código para una función Lambda. En este ejemplo, hay un perfil de confianza y las implementaciones se rechazan si las comprobaciones de firma fallan.

```
Resources:
  HelloWorld:
    Type: AWS::Serverless::Function
    Properties:
      CodeUri: hello_world/
      Handler: app.lambda_handler
      Runtime: python3.7
      CodeSigningConfigArn: !Ref MySignedFunctionCodeSigningConfig

  MySignedFunctionCodeSigningConfig:
    Type: AWS::Lambda::CodeSigningConfig
    Properties:
```



```
Description: "Code Signing for MySignedLambdaFunction"
AllowedPublishers:
  SigningProfileVersionArns:
    - MySigningProfile-profileVersionArn
CodeSigningPolicies:
  UntrustedArtifactOnDeployment: "Enforce"
```

Firmar tu código

Puedes firmar el código al empaquetar o implementar la aplicación. Especifique la opción `--signing-profiles` con el comando `sam package` o `sam deploy`, como se muestra en los siguientes comandos de ejemplo.

Firmar el código de función al empaquetar la aplicación:

```
sam package --signing-profiles HelloWorld=MySigningProfile --s3-bucket amzn-s3-demo-bucket --output-template-file packaged.yaml
```

Firmar el código de la función y la capa de la que depende la función al empaquetar la aplicación:

```
sam package --signing-profiles HelloWorld=MySigningProfile MyLayer=MySigningProfile --s3-bucket amzn-s3-demo-bucket --output-template-file packaged.yaml
```

Firmar el código de la función y una capa y, a continuación, realizar una implementación:

```
sam deploy --signing-profiles HelloWorld=MySigningProfile MyLayer=MySigningProfile --s3-bucket amzn-s3-demo-bucket --template-file packaged.yaml --stack-name --region us-east-1 --capabilities CAPABILITY_IAM
```

Note

Para poder firmar correctamente el código con los comandos `sam package` o `sam deploy`, el control de versiones debes estar habilitado para el bucket de Amazon S3 que utiliza con estos comandos. Si utiliza el bucket de Amazon S3 que se AWS SAM crea para usted, el control de versiones se habilita automáticamente. Para obtener más información sobre el control de versiones de los buckets de Amazon S3 e instrucciones para habilitar el control de versiones en un bucket de Amazon S3 que proporciones, consulta [Uso del control de versiones en los buckets de Amazon S3](#) en la Guía del usuario de Amazon Simple Storage Service.

Proporcionar perfiles de firma con **sam deploy --guided**

Cuando ejecuta el `sam deploy --guided` comando con una aplicación sin servidor configurada con firma de código, AWS SAM le pide que proporcione el perfil de firma que se utilizará para la firma de código. Para obtener más información sobre `sam deploy --guided`, consulta [sam deploy](#) en la referencia de los comandos de AWS SAM CLI.

Validar archivos AWS SAM de plantilla

Valide sus plantillas con [sam validate](#). Actualmente, este comando valida que la plantilla proporcionada sea JSON/YAML válida. Como ocurre con la mayoría de comandos de la CLI de AWS SAM, busca un archivo `template.[yaml|yml]` en el directorio de trabajo actual de forma predeterminada. Puede especificar una plantilla diferente `file/location` con la `--template` opción `-t o`.

Ejemplo:

```
$ sam validate
<path-to-template>/template.yaml is a valid SAM Template
```

Note

El `sam validate` comando requiere que se configuren AWS las credenciales. Para obtener más información, consulte [Configuración de la AWS SAM de CLI](#).

Cree su aplicación con AWS SAM

Tras añadir la infraestructura como código (IaC) a la AWS SAM plantilla, estará listo para empezar a crear la aplicación mediante el `sam build` comando. Este comando crea artefactos de compilación a partir de los archivos del directorio de proyectos de la aplicación (es decir, el archivo de AWS SAM plantilla, el código de la aplicación y cualquier archivo y dependencia aplicable específico del idioma). Estos artefactos de compilación preparan la aplicación sin servidor para las etapas posteriores del desarrollo de la aplicación, como las pruebas locales y la implementación en la nube. AWS Tanto las pruebas como la implementación utilizan artefactos de compilación como entradas.

Puede utilizar `sam build` para compilar toda la aplicación sin servidor. Además, puede crear compilaciones personalizadas, por ejemplo, con funciones, capas o tiempos de ejecución

personalizados específicos. Para obtener más información sobre cómo y por qué se usa `sam build`, consulte los temas de esta sección. Si deseas una introducción sobre el uso del comando `sam build`, consulta [Introducción a la construcción con AWS SAM](#).

Temas

- [Introducción a la construcción con AWS SAM](#)
- [Compilación predeterminada con AWS SAM](#)
- [Personaliza las compilaciones con AWS SAM](#)

Introducción a la construcción con AWS SAM

Utilice el AWS Serverless Application Model comando Command Line Interface (AWS SAMCLI) `sam build` para preparar la aplicación sin servidor para los siguientes pasos del flujo de trabajo de desarrollo, como las pruebas locales o la Nube de AWS implementación en. Este comando crea un directorio `.aws-sam` que estructura la aplicación en el formato y la ubicación que necesite `sam local` `sam deploy`.

- Para obtener una introducción al AWS SAMCLI, consulte [¿Qué es el AWS SAM CLI?](#).
- Para ver una lista de opciones de comandos `sam build`, consulta [compilación de sam](#).
- Para ver un ejemplo de uso de `sam build` durante un flujo de trabajo de desarrollo típico, consulta [Paso 2: crear tu propia aplicación](#).

Note

El uso de `sam build` requiere que comience con los componentes básicos de una aplicación sin servidor en su máquina de desarrollo. Esto incluye una AWS SAM plantilla, un código de función de AWS Lambda evento y cualquier archivo y dependencia específicos del idioma. Para obtener más información, consulte [Crea tu aplicación en AWS SAM](#).

Temas

- [Creación de aplicaciones con la misma compilación](#)
- [Prueba e implementación locales](#)
- [Prácticas recomendadas](#)

- [Opciones para la compilación de Sam](#)
- [Resolución de problemas](#)
- [Ejemplos](#)
- [Más información](#)

Creación de aplicaciones con la misma compilación

Antes de usar `sam build`, considere la posibilidad de configurar lo siguiente:

1. Funciones y capas Lambda: el comando `sam build` puede crear funciones y capas Lambda. Para obtener más información sobre el uso de las capas de Lambda, consulta [Creación de capas Lambda en AWS SAM](#).
2. Tiempo de ejecución de Lambda: el tiempo de ejecución proporciona un entorno específico del lenguaje que ejecuta su función en un entorno de ejecución cuando se invoca. Puede configurar tiempos de ejecución nativos y personalizados.
 - a. Tiempo de ejecución nativo: cree sus funciones de Lambda en un tiempo de ejecución de Lambda compatible y cree sus funciones para utilizar un tiempo de ejecución de Lambda nativo en Nube de AWS
 - b. Tiempo de ejecución personalizado: cree sus funciones de Lambda con cualquier lenguaje de programación y cree su tiempo de ejecución mediante un proceso personalizado definido en un generador makefile o de terceros, como esbuild. Para obtener más información, consulta [Creación de funciones de eventos de Lambda con tiempos de ejecución personalizados en AWS SAM](#).
3. Tipo de paquete Lambda: las funciones de Lambda se pueden empaquetar en los siguientes tipos de paquetes de implementación de Lambda:
 - a. Un archivo de archivo .zip, que incluye el código de la aplicación y sus dependencias.
 - b. Imagen del contenedor: contiene el sistema operativo base, el tiempo de ejecución, las extensiones de Lambda, el código de tu aplicación y sus dependencias.

Estos ajustes de la aplicación se pueden configurar al inicializar una aplicación mediante `sam init`

- Para obtener más información acerca del uso de `sam init`, consulta [Crea tu aplicación en AWS SAM](#).
- Para obtener más información sobre cómo configurar estos ajustes en la aplicación, consulta [Compilación predeterminada con AWS SAM](#).

Para compilar una aplicación

1. `cd` a la raíz de su proyecto. Se trata de la misma ubicación que la plantilla. AWS SAM

```
$ cd sam-app
```

2. Ejecuta lo siguiente:

```
sam-app $ sam build <arguments> <options>
```

Note

Una opción que se usa con frecuencia es `--use-container`. Para obtener más información, consulta [Creación de una función de eventos Lambda dentro de un contenedor proporcionado](#).

A continuación, se muestra un ejemplo de resultados del AWS SAM CLI output.

```
sam-app $ sam build
Starting Build use cache
Manifest file is changed (new hash: 3298f1304...d4d421) or dependency folder (.aws-sam/deps/4d3dfad6-a267-47a6-a6cd-e07d6fae318c) is missing for (HelloWorldFunction),
downloading dependencies and copying/building source
Building codeuri: /Users/.../sam-app/hello_world runtime: python3.12 metadata: {}
architecture: x86_64 functions: HelloWorldFunction
Running PythonPipBuilder:Cleanup
Running PythonPipBuilder:ResolveDependencies
Running PythonPipBuilder:CopySource
Running PythonPipBuilder:CopySource

Build Succeeded

Built Artifacts  : .aws-sam/build
Built Template   : .aws-sam/build/template.yaml

Commands you can use next
=====
[*] Validate SAM template: sam validate
[*] Invoke Function: sam local invoke
[*] Test Function in the Cloud: sam sync --stack-name {{stack-name}} --watch
```

```
[*] Deploy: sam deploy --guided
```

3. La CLI de AWS SAM crea un directorio de compilación `.aws-sam`. A continuación, se muestra un ejemplo:

```
.aws-sam
### build
#   ### HelloWorldFunction
#   #   ### __init__.py
#   #   ### app.py
#   #   ### requirements.txt
#   ### template.yaml
### build.toml
```

Según cómo esté configurada la aplicación, la CLI de AWS SAM hace lo siguiente:

1. Descarga, instala y organiza las dependencias en el directorio `.aws-sam/build`.
2. Prepara el código Lambda. Esto puede incluir la compilación del código, la creación de binarios ejecutables y la creación de imágenes de contenedores.
3. Copia los artefactos de construcción en el directorio `.aws-sam`. El formato variará en función del tipo de paquete de la aplicación.
 - a. En el caso de los tipos de paquetes `zip`, los artefactos aún no están comprimidos, por lo que se pueden utilizar en testeos locales. La CLI de AWS SAM comprime la aplicación cuando se usa `sam deploy`.
 - b. Para los tipos de paquetes de imágenes de contenedores, se crea una imagen de contenedor localmente y se hace referencia a ella en el archivo `.aws-sam/build.toml`.
4. Copia la AWS SAM plantilla en el `.aws-sam` directorio y la modifica con nuevas rutas de archivo cuando es necesario.

Los componentes principales que componen los artefactos de compilación del directorio `.aws-sam` son los siguientes:

- El directorio de compilación: contiene las funciones y capas de Lambda estructuradas de forma independiente. Esto da como resultado una estructura única para cada función o capa del directorio `.aws-sam/build`.
- La AWS SAM plantilla: se modifica con valores actualizados en función de los cambios ocurridos durante el proceso de creación.

- El archivo `build.toml`: un archivo de configuración que contiene los ajustes de compilación utilizados por `AWS SAMCLI`

Prueba e implementación locales

Al realizar testeo local con `sam local` o realizar una implementación con `sam deploy`, la CLI de AWS SAM hace lo siguiente:

1. Primero comprueba si existe un `.aws-sam` directorio y si hay una AWS SAM plantilla ubicada dentro de ese directorio. Si se cumplen estas condiciones, la CLI de AWS SAM considera el directorio raíz de la aplicación.
2. Si no se cumplen estas condiciones, AWS SAMCLI considera la ubicación original de la AWS SAM plantilla como el directorio raíz de la aplicación.

Durante el desarrollo, si se realizan cambios en los archivos originales de la aplicación, ejecuta `sam build` para actualizar el directorio `.aws-sam` antes de realizar testeo local.

Prácticas recomendadas

- No edite ningún código del directorio `.aws-sam/build`. En su lugar, actualice el código fuente original en la carpeta del proyecto y ejecuta `sam build` para actualizar el directorio `.aws-sam/build`.
- Cuando modifique los archivos originales, ejecuta `sam build` para actualizar el directorio `.aws-sam/build`.
- Es posible que desee que la CLI de AWS SAM haga referencia al directorio raíz original del proyecto en lugar de al directorio `.aws-sam`, por ejemplo, al desarrollar y probar con `sam local`. Elimine el `.aws-sam` directorio o la AWS SAM plantilla del `.aws-sam` directorio para que AWS SAMCLI reconozcan el directorio del proyecto original como el directorio raíz del proyecto. Cuando esté listo, ejecuta nuevamente `sam build` para crear el directorio `.aws-sam`.
- Cuando se ejecuta `sam build`, el `.aws-sam/build` directorio se sobrescribe cada vez. El directorio `.aws-sam` no lo hace. Si desea almacenar archivos como registros, guárdalos en `.aws-sam` para evitar que se sobrescriban.

Opciones para la compilación de Sam

Creación de un solo recurso

Proporciona el identificador lógico del recurso para crear solo ese recurso. A continuación, se muestra un ejemplo:

```
$ sam build HelloWorldFunction
```

Para crear un recurso a partir de una aplicación o pila anidada, proporciona el identificador lógico de la aplicación o pila junto con el identificador lógico del recurso con el siguiente formato *<stack-logical-id>/<resource-logical-id>*:

```
$ sam build MyNestedStack/MyFunction
```

Creación de una función de eventos Lambda dentro de un contenedor proporcionado

La opción `--use-container` descarga una imagen de contenedor y la usa para crear las funciones de Lambda. A continuación, se hace referencia al contenedor local en el archivo `.aws-sam/build.toml`.

Esto requiere que Docker esté instalado. Para obtener instrucciones, consulta [Instalar Docker](#).

A continuación, se muestra un ejemplo del comando:

```
$ sam build --use-container
```

Puedes especificar la imagen del contenedor que se usará con la opción `--build-image`. A continuación, se muestra un ejemplo:

```
$ sam build --use-container --build-image amazon/aws-sam-cli-build-image-nodejs20.x
```

Para especificar la imagen del contenedor que se usará para una sola función, proporciona el ID lógico de la función. A continuación, se muestra un ejemplo:

```
$ sam build --use-container --build-image Function1=amazon/aws-sam-cli-build-image-python3.12
```


Creación de imágenes de contenedores con BuildKit

Al crear funciones Lambda con el tipo de Image paquete, puede utilizarlas BuildKit como motor de compilación pasando la `--use-buildkit` opción. BuildKit proporciona un rendimiento de compilación mejorado, un mejor almacenamiento en caché y un resultado de compilación mejorado en comparación con el motor de Docker compilación anterior.

Esta opción requiere tener el Docker o Finch CLIs instalado y en suPATH.

Puede combinarla `--use-buildkit` con otras opciones de construcción. A continuación, se muestra un ejemplo:

```
$ sam build --use-buildkit --parallel
```

Note

El `--use-buildkit` indicador es solo para compilaciones de imágenes de contenedores. No tendrá ningún efecto en las funciones zip con las que se compilen `--use-container`.

Transfiera la variables de entorno al contenedor de compilaciones.

Utiliza `--container-env-var` para transferir las variables de entorno al contenedor de compilaciones. A continuación, se muestra un ejemplo:

```
$ sam build --use-container --container-env-var Function1.GITHUB_TOKEN=<token1> --  
container-env-var GLOBAL_ENV_VAR=<global-token>
```

Para transferir variables de entorno de un archivo, utiliza la opción `--container-env-var-file`. A continuación, se muestra un ejemplo:

```
$ sam build --use-container --container-env-var-file <env.json>
```

Ejemplo del archivo `env.json`:

```
{  
  "MyFunction1": {  
    "GITHUB_TOKEN": "TOKEN1"  
  },  
  "MyFunction2": {
```

```
"GITHUB_TOKEN": "TOKEN2"  
}  
}
```

Acelere la creación de aplicaciones que contienen múltiples funciones

Cuando se ejecuta `sam build` en una aplicación con varias funciones, la CLI de AWS SAM crea cada función de una en una. Para acelerar el proceso de compilación, utiliza la opción `--parallel`. De esta forma, se crean todas las funciones y capas al mismo tiempo.

A continuación, se muestra un ejemplo del comando:

```
$ sam build --parallel
```

Acelere los tiempos de compilación al compilar el proyecto en la carpeta de origen

Para consultar los tiempos de ejecución y los métodos de compilación compatibles, puedes usar la opción de `--build-in-source` para compilar el proyecto directamente en la carpeta de origen. De forma predeterminada, se AWS SAM CLI compila en un directorio temporal, lo que implica copiar el código fuente y los archivos del proyecto. Con `--build-in-source` las AWS SAM CLI compilaciones directamente en la carpeta de origen, lo que acelera el proceso de compilación al eliminar la necesidad de copiar los archivos a un directorio temporal.

Para obtener una lista de los tiempos de ejecución y métodos de compilación compatibles, consulta [--build-in-source](#).

Resolución de problemas

Para solucionar el problema AWS SAMCLI, consulte [AWS SAM CLI solución de problemas](#).

Ejemplos

Creación de una aplicación que utiliza un tiempo de ejecución nativo y un tipo de paquete .zip

Para este ejemplo, consulta [Tutorial: Implemente una aplicación Hello World con AWS SAM](#).

Crear una aplicación que utiliza un tiempo de ejecución y un tipo de paquete de imágenes nativas

Primero, ejecuta `sam init` para inicializar una nueva aplicación. Durante el flujo interactivo, seleccione el tipo de paquete Image. A continuación, se muestra un ejemplo:

```
$ sam init
```

...

Which template source would you like to use?

- 1 - AWS Quick Start Templates
- 2 - Custom Template Location

Choice: **1**

Choose an AWS Quick Start application template

- 1 - Hello World Example
- 2 - Multi-step workflow
- 3 - Serverless API
- 4 - Scheduled task
- 5 - Standalone function
- 6 - Data processing
- 7 - Hello World Example With Powertools
- 8 - Infrastructure event management
- 9 - Serverless Connector Hello World Example
- 10 - Multi-step workflow with Connectors
- 11 - Lambda EFS example
- 12 - DynamoDB Example
- 13 - Machine Learning

Template: **1**

Use the most popular runtime and package type? (Python and zip) [y/N]: **ENTER**

Which runtime would you like to use?

...

- 10 - java8
- 11 - nodejs20.x
- 12 - nodejs18.x
- 13 - nodejs16.x

...

Runtime: **12**

What package type would you like to use?

- 1 - Zip
- 2 - Image

Package type: **2**

Based on your selections, the only dependency manager available is npm.
We will proceed copying the template using npm.

Would you like to enable X-Ray tracing on the function(s) in your application? [y/N]: **ENTER**

Would you like to enable monitoring using CloudWatch Application Insights?
 For more info, please view <https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/cloudwatch-application-insights.html> [y/N]: *ENTER*

Project name [sam-app]: *ENTER*

Cloning from <https://github.com/aws/aws-sam-cli-app-templates> (process may take a moment)

```
-----
Generating application:
-----
Name: sam-app
Base Image: amazon/nodejs18.x-base
Architectures: x86_64
Dependency Manager: npm
Output Directory: .
Configuration file: sam-app/samconfig.toml
```

Next steps can be found in the README file at sam-app/README.md

...

AWS SAMCLI inicializa una aplicación y crea el siguiente directorio de proyectos:

```
sam-app
### README.md
### events
#   ### event.json
### hello-world
#   ### Dockerfile
#   ### app.mjs
#   ### package.json
#   ### tests
#       ### unit
#           ### test-handler.mjs
### samconfig.toml
### template.yaml
```

A continuación, ejecuta `sam build` para crear e implementar nuestra aplicación:

```
sam-app $ sam build
```

```

Building codeuri: /Users/.../build-demo/sam-app runtime: None metadata: {'DockerTag':
'nodejs18.x-v1', 'DockerContext': '/Users/.../build-demo/sam-app/hello-world',
'Dockerfile': 'Dockerfile'} architecture: arm64 functions: HelloWorldFunction
Building image for HelloWorldFunction function
Setting DockerBuildArgs: {} for HelloWorldFunction function
Step 1/4 : FROM public.ecr.aws/lambda/nodejs:18
---> f5b68038c080
Step 2/4 : COPY app.mjs package*.json ./
---> Using cache
---> 834e565aae80
Step 3/4 : RUN npm install
---> Using cache
---> 31c2209dd7b5
Step 4/4 : CMD ["app.lambdaHandler"]
---> Using cache
---> 2ce2a438e89d
Successfully built 2ce2a438e89d
Successfully tagged helloworldfunction:nodejs18.x-v1

Build Succeeded

Built Artifacts  : .aws-sam/build
Built Template   : .aws-sam/build/template.yaml

Commands you can use next
=====
[*] Validate SAM template: sam validate
[*] Invoke Function: sam local invoke
[*] Test Function in the Cloud: sam sync --stack-name {{stack-name}} --watch
[*] Deploy: sam deploy --guided

```

Crear una aplicación que incluya un lenguaje de programación compilado

En este ejemplo, creamos una aplicación que contiene una función de Lambda utilizando el tiempo de ejecución Go.

En primer lugar, inicializamos una nueva aplicación utilizando `sam init` y configuramos nuestra aplicación para que utiliza Go:

```
$ sam init
```

```
...
```

Which template source would you like to use?

- 1 - AWS Quick Start Templates
- 2 - Custom Template Location

Choice: **1**

Choose an AWS Quick Start application template

- 1 - Hello World Example
- 2 - Multi-step workflow
- 3 - Serverless API

...

Template: **1**

Use the most popular runtime and package type? (Python and zip) [y/N]: **ENTER**

Which runtime would you like to use?

...

- 4 - dotnetcore3.1
- 5 - go1.x
- 6 - go (provided.al2)

...

Runtime: **5**

What package type would you like to use?

- 1 - Zip
- 2 - Image

Package type: **1**

Based on your selections, the only dependency manager available is mod.
We will proceed copying the template using mod.

Would you like to enable X-Ray tracing on the function(s) in your application? [y/N]: **ENTER**

Would you like to enable monitoring using CloudWatch Application Insights?
For more info, please view <https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/cloudwatch-application-insights.html> [y/N]: **ENTER**

Project name [sam-app]: **ENTER**

Cloning from <https://github.com/aws/aws-sam-cli-app-templates> (process may take a moment)

Generating application:

```

-----
Name: sam-app
Runtime: go1.x
Architectures: x86_64
Dependency Manager: mod
Application Template: hello-world
Output Directory: .
Configuration file: sam-app/samconfig.toml

Next steps can be found in the README file at sam-app-go/README.md

```

...

AWS SAMCLI inicializa la aplicación. A continuación, se muestra la estructura de directorios que utilizamos para esta aplicación de ejemplo:

```

sam-app
### Makefile
### README.md
### events
#   ### event.json
### hello-world
#   ### go.mod
#   ### go.sum
#   ### main.go
#   ### main_test.go
### samconfig.toml
### template.yaml

```

Hacemos referencia al archivo `README.md` para conocer los requisitos de esta aplicación.

```

...
## Requirements
* AWS CLI already configured with Administrator permission
* [Docker installed](https://www.docker.com/community-edition)
* [Golang](https://golang.org)
* SAM CLI - [Install the SAM CLI](https://docs.aws.amazon.com/serverless-application-model/latest/developerguide/serverless-sam-cli-install.html)
...

```

A continuación, ejecutamos `sam local invoke` para probar la función. Este comando genera errores, ya que Go no está instalado en nuestra máquina local:

```

sam-app $ sam local invoke
Invoking hello-world (go1.x)
Local image was not found.
Removing rapid images for repo public.ecr.aws/sam/emulation-go1.x
Building
  image.....
Using local image: public.ecr.aws/lambda/go:1-rapid-x86_64.

Mounting /Users/.../Playground/build/sam-app/hello-world as /var/task:ro,delegated
  inside runtime container
START RequestId: c6c5eddf-042b-4e1e-ba66-745f7c86dd31 Version: $LATEST
fork/exec /var/task/hello-world: no such file or directory: PathError
null
END RequestId: c6c5eddf-042b-4e1e-ba66-745f7c86dd31
REPORT RequestId: c6c5eddf-042b-4e1e-ba66-745f7c86dd31  Init Duration: 0.88 ms
  Duration: 175.75 ms Billed Duration: 176 ms Memory Size: 128 MB      Max Memory Used:
  128 MB
{"errorMessage":"fork/exec /var/task/hello-world: no such file or
  directory","errorType":"PathError"}%

```

A continuación, ejecuta `sam build` para crear e implementar la aplicación. Nos encontramos con un error, ya que Go no está instalado en nuestra máquina local:

```

sam-app $ sam build
Starting Build use cache
Cache is invalid, running build and copying resources for following functions
  (HelloWorldFunction)
Building codeuri: /Users/.../Playground/build/sam-app/hello-world runtime: go1.x
  metadata: {} architecture: x86_64 functions: HelloWorldFunction

Build Failed
Error: GoModulesBuilder:Resolver - Path resolution for runtime: go1.x of binary: go was
  not successful

```

Si bien podríamos configurar nuestra máquina local para que construya correctamente nuestra función, en su lugar utilizamos la opción `--use-container` con `sam build`. AWS SAMCLI Descarga una imagen de contenedor, crea nuestra función con la nativa `GoModulesBuilder` y copia el binario resultante en nuestro `.aws-sam/build/HelloWorldFunction` directorio.

```

sam-app $ sam build --use-container
Starting Build use cache

```



```

Starting Build inside a container
Cache is invalid, running build and copying resources for following functions
(HelloWorldFunction)
Building codeuri: /Users/.../build/sam-app/hello-world runtime: go1.x metadata: {}
architecture: x86_64 functions: HelloWorldFunction

Fetching public.ecr.aws/sam/build-go1.x:latest-x86_64 Docker container
image.....
Mounting /Users/.../build/sam-app/hello-world as /tmp/samcli/source:ro,delegated inside
runtime container
Running GoModulesBuilder:Build

Build Succeeded

Built Artifacts   : .aws-sam/build
Built Template    : .aws-sam/build/template.yaml

Commands you can use next
=====
[*] Validate SAM template: sam validate
[*] Invoke Function: sam local invoke
[*] Test Function in the Cloud: sam sync --stack-name {{stack-name}} --watch
[*] Deploy: sam deploy --guided

```

A continuación, se muestra un ejemplo del archivo `.aws-sam`.

```

.aws-sam
### build
#   ### HelloWorldFunction
#   #   ### hello-world
#   ### template.yaml
### build.toml
### cache
#   ### c860d011-4147-4010-addb-2eaa289f4d95
#       ### hello-world
### deps

```

A continuación, ejecutamos `sam local invoke`. Nuestra función se invocó correctamente:

```

sam-app $ sam local invoke
Invoking hello-world (go1.x)
Local image is up-to-date
Using local image: public.ecr.aws/lambda/go:1-rapid-x86_64.

```

```
Mounting /Users/.../Playground/build/sam-app/.aws-sam/build/HelloWorldFunction as /var/
task:ro,delegated inside runtime container
START RequestId: cfc8ffa8-29f2-49d4-b461-45e8c7c80479 Version: $LATEST
END RequestId: cfc8ffa8-29f2-49d4-b461-45e8c7c80479
REPORT RequestId: cfc8ffa8-29f2-49d4-b461-45e8c7c80479  Init Duration: 1.20 ms
      Duration: 1782.46 ms      Billed Duration: 1783 ms      Memory Size: 128 MB
      Max Memory Used: 128 MB
{"statusCode":200,"headers":null,"multiValueHeaders":null,"body":"Hello,
72.21.198.67\n"}%
```

Más información

Para obtener más información sobre el uso del comando `sam build`, consulta los enlaces siguientes:

- [Aprendizaje AWS SAM: `sam build`](#) — Serverless Land comienza con la serie AWS SAM«Learning». YouTube
- [Aprendizaje AWS SAM | `sam build` | E3](#) — Serie «Aprendizaje AWS SAM» de Serverless Land en adelante. YouTube
- [AWS SAM build: cómo proporciona artefactos para su implementación \(Sesiones con SAM S2E8\) — Sesiones](#) con series posteriores. AWS SAM YouTube
- [AWS SAM compilaciones personalizadas: Cómo usar Makefiles para personalizar compilaciones en SAM \(S2E9\) — Sesiones](#) con series posteriores. AWS SAM YouTube

Compilación predeterminada con AWS SAM

Para crear una aplicación sin servidor, utiliza el comando [compilación de sam](#). Este comando también recopila los artefactos de compilación de las dependencias de la aplicación y los coloca en el formato y la ubicación adecuados para los siguientes pasos, como las pruebas, el empaquetado y la implementación locales.

Las dependencias de la aplicación se especifican en un archivo de manifiesto, como `requirements.txt` (Python) o `package.json` (Node.js), o mediante la `Layers` propiedad de un recurso de función. La `Layers` propiedad contiene una lista de los recursos de [AWS Lambda capa](#) de los que depende la función de eventos de Lambda.

El formato de los artefactos de compilación de la aplicación depende de la propiedad `PackageType` de cada función. Las opciones de esta propiedad son:

- **Zip:** un archivo .zip, que incluye el código de la aplicación y sus dependencias. Si comprimes el código como un archivo .zip, debes especificar un tiempo de ejecución de Lambda para tu función.
- **Image:** una imagen del contenedor, que incluye el sistema operativo base, el tiempo de ejecución, las extensiones, el código de la aplicación y sus dependencias.

Para obtener más información sobre los tipos de paquetes de Lambda, consulta [Paquetes de implementación de Lambda](#) en la Guía para desarrolladores de AWS Lambda .

Temas

- [Creación de un archivo .zip](#)
- [Creación de una imagen del contenedor](#)
- [Archivo de variables del entorno del contenedor.](#)
- [Acelera los tiempos de compilación al compilar el proyecto en la carpeta de origen](#)
- [Ejemplos](#)
- [Crear funciones fuera de AWS SAM](#)

Creación de un archivo .zip

Para crear una aplicación sin servidor como un archivo .zip, declare `PackageType: Zip` para tu función sin servidor.

AWS SAM crea la aplicación para la [arquitectura](#) que especifique. Si no especifica una arquitectura, la AWS SAM utiliza de forma `x86_64` predeterminada.

Si la función de Lambda depende de paquetes que tienen programas compilados de forma nativa, utiliza el indicador `--use-container`. Este indicador compila tus funciones de forma local en un contenedor que se comporta como un entorno Lambda, de modo que estén en el formato correcto cuando las despliegues en la nube. AWS

Cuando utiliza `--use-container` esta opción, AWS SAM extrae de forma predeterminada la imagen del contenedor de [Amazon ECR Public](#). Si desea extraer una imagen de contenedor de otro repositorio o de una versión específica de AWS SAM CLI, puede usar la `--build-image` opción y proporcionar el URI de una imagen de contenedor alternativa. A continuación, se muestran dos comandos de ejemplo para crear aplicaciones mediante imágenes de contenedor de una versión específica de AWS SAM CLI:

```
# Build a Node.js 20 application using a container image for a specific version of AWS
SAM CLI (1.136.0)
sam build --use-container --build-image public.ecr.aws/sam/build-nodejs22.x:1.136.0

# Build a function resource using the Python 3.13 container image from a specific
version of AWS SAM CLI (1.136.0)(
sam build --use-container --build-image Function1=public.ecr.aws/sam/build-
python3.13:1.136.0
```

Para ver ejemplos adicionales de cómo crear una aplicación de archivo .zip, consulta la sección de ejemplos que aparece más adelante en este tema.

Creación de una imagen del contenedor

Para crear una aplicación sin servidor como una imagen del contenedor, declara `PackageType: Image` para tu función sin servidor. También debes declarar el atributo de recurso `Metadata` con las siguientes entradas:

`Dockerfile`

El nombre de `Dockerfile` que se asocia con la función de Lambda.

`DockerContext`

La ubicación del `Dockerfile`.

`DockerTag`

(Opcional) Una etiqueta para aplicar a la imagen creada.

`DockerBuildArgs`

Crear argumentos para la compilación.

Important

La CLI de AWS SAM no elimina ni ofusca ninguna información que incluya en los argumentos `DockerBuildArgs`. Recomendamos encarecidamente que no utilice esta sección para almacenar información confidencial, como contraseñas o secretos.

DockerImage

(Opcional) Un nombre personalizado para la imagen de Docker creada. Si no se especifica, se utiliza el ID de recurso lógico de la función.

A continuación, se muestra un ejemplo de una sección de atributos de recursos de Metadata:

```
Metadata:
  Dockerfile: Dockerfile
  DockerContext: ./hello_world
  DockerTag: v1
  DockerImage: my-custom-image
```

Para descargar una aplicación de muestra configurada con el tipo de paquete Image, consulte [Tutorial: Implemente una aplicación Hello World con AWS SAM](#). Cuando aparezca el mensaje que le pregunte qué tipo de paquete quieres instalar, elige Image.

Note

Si especificas una imagen base multiarquitectura en tu Dockerfile, AWS SAM crea tu imagen de contenedor para la arquitectura de tu máquina host. Para crear una arquitectura diferente, especifique una imagen base que utiliza la arquitectura de destino específica.

De forma predeterminada, AWS SAMCLI utiliza el motor de Docker compilación anterior al crear imágenes de contenedores. Para utilizar BuildKit un backend de contenedores mejorado que mejore el rendimiento y gestione compilaciones más complejas, basta con `--use-buildkit` usar el comando de compilación:

```
$ sam build --use-buildkit
```

El uso `--use-buildkit` requiere tener Finch CLI instalado Docker o.

Archivo de variables del entorno del contenedor.

Para proporcionar un archivo JSON que contenga variables de entorno para el contenedor de compilación, usa el argumento `--container-env-var-file` junto con el comando `sam build`. Puedes proporcionar una única variable de entorno que se aplique a todos los recursos sin servidor o variables de entorno diferentes para cada recurso.

Formato

El formato para pasar las variables de entorno a un contenedor de compilación depende del número de variables de entorno que proporciones para tus recursos.

Para proporcionar una única variable de entorno para todos los recursos, especifique un objeto `Parameters` como el siguiente:

```
{
  "Parameters": {
    "GITHUB_TOKEN": "TOKEN_GLOBAL"
  }
}
```

Para proporcionar variables de entorno diferentes para cada recurso, especifique objetos para cada recurso de la siguiente manera:

```
{
  "MyFunction1": {
    "GITHUB_TOKEN": "TOKEN1"
  },
  "MyFunction2": {
    "GITHUB_TOKEN": "TOKEN2"
  }
}
```

Guarda las variables de entorno como un archivo, por ejemplo, denominado `env.json`. El siguiente comando usa este archivo para pasar las variables de entorno al contenedor de compilación:

```
sam build --use-container --container-env-var-file env.json
```

Prioridad

- Las variables de entorno que proporcionas para recursos específicos tienen prioridad sobre la variable de entorno única para todos los recursos.
- Las variables de entorno que se proporcionan en la línea de comandos tienen prioridad sobre las variables de entorno de un archivo.

Acelera los tiempos de compilación al compilar el proyecto en la carpeta de origen

Para consultar los tiempos de ejecución y los métodos de compilación compatibles, puedes usar la opción de `--build-in-source` para compilar el proyecto directamente en la carpeta de origen. De forma predeterminada, se AWS SAM CLI compila en un directorio temporal, lo que implica copiar el código fuente y los archivos del proyecto. Con `--build-in-source` las AWS SAM CLI compilaciones directamente en la carpeta de origen, lo que acelera el proceso de compilación al eliminar la necesidad de copiar los archivos a un directorio temporal.

Para consultar una lista de los tiempos de ejecución y métodos de compilación admitidos, consulta [--build-in-source](#).

Ejemplos

Ejemplo 1: Archivo .zip

Los siguientes comandos `sam build` crean un archivo .zip:

```
# Build all functions and layers, and their dependencies
sam build

# Run the build process inside a Docker container that functions like a Lambda
environment
sam build --use-container

# Build a Node.js 20 application using a container image for a specific version of AWS
SAM CLI (1.136.0)
sam build --use-container --build-image public.ecr.aws/sam/build-nodejs22.x:1.136.0

# Build a function resource using the Python 3.13 container image from a specific
version of AWS SAM CLI (1.136.0)(
sam build --use-container --build-image Function1=public.ecr.aws/sam/build-
python3.13:1.136.0

# Build and run your functions locally
sam build && sam local invoke

# For more options
sam build --help
```

Ejemplo 2: Imagen del contenedor

La siguiente AWS SAM plantilla se compila como una imagen contenedora:

```
Resources:
  HelloWorldFunction:
    Type: AWS::Serverless::Function
    Properties:
      PackageType: Image
      ImageConfig:
        Command: ["app.lambda_handler"]
    Metadata:
      Dockerfile: Dockerfile
      DockerContext: ./hello_world
      DockerTag: v1
```

A continuación, se muestra un ejemplo de un Dockerfile.

```
FROM public.ecr.aws/lambda/python:3.12

COPY app.py requirements.txt ./

RUN python3.12 -m pip install -r requirements.txt

# Overwrite the command by providing a different command directly in the template.
CMD ["app.lambda_handler"]
```

Ejemplo 3: npm ci

En el Node.js caso de las aplicaciones, puede `npm ci` utilizarlas en lugar de `npm install` instalarlas. Para usar `npm ci`, especifique `UseNpmCi: True` bajo `BuildProperties` en el atributo de recurso `Metadata` de la función de Lambda. Para poder utilizar `npm ci`, la aplicación debe tener un archivo `package-lock.json` o `npm-shrinkwrap.json` presente en `CodeUri` para la función de Lambda.

El siguiente ejemplo utiliza `npm ci` para instalar dependencias cuando ejecuta `sam build`:

```
Resources:
  HelloWorldFunction:
    Type: AWS::Serverless::Function
    Properties:
      CodeUri: hello-world/
      Handler: app.handler
      Runtime: nodejs20.x
      Architectures:
```



```
- x86_64
Events:
  HelloWorld:
    Type: Api
    Properties:
      Path: /hello
      Method: get
Metadata:
  BuildProperties:
    UseNpmCi: True
```

Paquetes principales de Python

En el caso de las aplicaciones de Python, puede conservar la estructura del paquete durante el proceso de compilación para permitir las importaciones absolutas. Para conservar la estructura del paquete, especifique `ParentPackageMode` en `BuildProperties` en el atributo de recurso `Metadata` de la función de Lambda.

El siguiente ejemplo conserva la estructura del paquete `app` cuando se ejecuta `sam build`:

```
Resources:
  HelloWorldFunction:
    Type: AWS::Serverless::Function
    Properties:
      CodeUri: hello-world/
      Handler: app.main.handler
      Runtime: python3.12
      Architectures:
        - x86_64
    Metadata:
      BuildProperties:
        ParentPackageMode: explicit
        ParentPackages: app
```

Con esta configuración, el código puede usar importaciones absolutas como `from app.utils import logger`, en lugar de importaciones relativas como `from .utils import logger`.

Crear funciones fuera de AWS SAM

De forma predeterminada, cuando se ejecuta `sam build`, AWS SAM se crean todos los recursos de la función. Otras opciones incluyen:

- Cree todos los recursos de funciones fuera de AWS SAM: si crea todos los recursos de funciones manualmente o mediante otra herramienta, no `sam build` es obligatorio. Puedes omitir `sam build` y pasar al siguiente paso del proceso, como realizar pruebas locales o implementar tu aplicación.
- Cree algunos recursos de funciones fuera de AWS SAM: si desea AWS SAM crear algunos de sus recursos de función y, al mismo tiempo, crear otros recursos de funciones fuera de AWS SAM ella, puede especificarlo en la AWS SAM plantilla.

Cree algunos recursos de funciones fuera de AWS SAM

Para AWS SAM omitir una función al utilizar `sam build`, configura lo siguiente en tu AWS SAM plantilla:

1. Añada la propiedad de metadatos `SkipBuild: True` a la función.
2. Especifique la ruta a los recursos de la función creada.

A continuación, se muestra un ejemplo, con `TestFunction` configurado para omitirse. Sus recursos creados se encuentran en `built-resources/TestFunction.zip`.

```
TestFunction:
  Type: AWS::Serverless::Function
  Properties:
    CodeUri: built-resources/TestFunction.zip
    Handler: TimeHandler::handleRequest
    Runtime: java11
  Metadata:
    SkipBuild: True
```

Ahora, cuando ejecutes `sam build`, AWS SAM hará lo siguiente:

1. AWS SAM omitirá las funciones configuradas con `SkipBuild: True`.
2. AWS SAM compilará todos los demás recursos de funciones y los guardará en caché en el directorio de `.aws-sam` compilación.
3. En el caso de las funciones omitidas, tu plantilla en el directorio de compilación `.aws-sam` se actualizará automáticamente para hacer referencia a la ruta especificada a los recursos de la función creada.

Este es un ejemplo de la plantilla almacenada en caché para `TestFunction` en el directorio de compilación `.aws-sam`:

```
TestFunction:
  Type: AWS::Serverless::Function
  Properties:
    CodeUri: ../../built-resources/TestFunction.zip
    Handler: TimeHandler::handleRequest
    Runtime: java11
  Metadata:
    SkipBuild: True
```

Personaliza las compilaciones con AWS SAM

Puede personalizar la compilación para incluir funciones de eventos de Lambda o capas de Lambda específicas. Una función es un recurso que puedes invocar para ejecutar el código en Lambda. Una capa de Lambda permite extraer código de una función de Lambda que luego se puede reutilizar en varias funciones de Lambda. Puede optar por personalizar su compilación con funciones de Lambda específicas si quiere centrarse en desarrollar e implementar funciones individuales sin servidor sin la complejidad de administrar dependencias o recursos compartidos. Además, puede optar por crear una capa de Lambda que le ayude a reducir el tamaño de los paquetes de implementación, a separar la lógica de las funciones principales de las dependencias y a compartir las dependencias entre varias funciones.

En los temas de esta sección se exploran algunas de las diferentes formas con las que se pueden crear funciones Lambda. AWS SAM Esto incluye la creación de funciones de Lambda con las versiones ejecutables de los clientes y la creación de capas de Lambda. Los tiempos de ejecución personalizados le permiten instalar y usar un lenguaje que no aparece en los tiempos de ejecución de Lambda en la Guía para desarrolladores. AWS Lambda Esto le permite crear un entorno de ejecución especializado para ejecutar funciones y aplicaciones sin servidor. Compilar solo capas de Lambda (en lugar de compilar toda la aplicación) puede beneficiarle de varias maneras. Puede ayudarle a reducir el tamaño de sus paquetes de implementación, separar la lógica de las funciones principales de las dependencias y permitirle compartir las dependencias entre varias funciones.

Para obtener más información, consulte [Conceptos de Lambda](#) en la Guía para desarrolladores de AWS Lambda .

Temas

- [Creación de funciones de eventos de Node.js Lambda con esbuild in AWS SAM](#)
- [Creación de funciones de eventos Lambda de .NET con compilación AOT nativa en AWS SAM](#)

- [Creación de funciones de eventos de Rust Lambda con Cargo Lambda in AWS SAM](#)
- [Creación de funciones de eventos Lambda de Python con uv in AWS SAM](#)
- [Creación de funciones de eventos de Lambda con tiempos de ejecución personalizados en AWS SAM](#)
- [Creación de capas Lambda en AWS SAM](#)

Creación de funciones de eventos de Node.js Lambda con esbuild in AWS SAM

Para crear y empaquetar funciones de Node.js AWS Lambda eventos, puede utilizarlas AWS SAMCLI con el paquete JavaScript esbuild. El paquete esbuild admite las funciones Lambda que usted escriba. TypeScript

Para crear una función Node.js Lambda con esbuild, añada un Metadata objeto a su `AWS::Serverless::Function` recurso y especifique esbuild el `BuildMethod`. Al ejecutar el `sam build` comando, AWS SAM usa esbuild para empaquetar el código de la función Lambda.

Propiedades de los metadatos

El objeto Metadata admite las siguientes propiedades.

BuildMethod

Especifica el paquete de URL para tu aplicación. El único valor admitido es `esbuild`.

BuildProperties

Especifica las propiedades de creación del código de la función de Lambda.

El objeto de `BuildProperties` admite las siguientes propiedades. Todas las propiedades son opcionales. De forma predeterminada, AWS SAM utiliza el controlador de funciones Lambda como punto de entrada.

EntryPoints

Especifica puntos de entrada para tu aplicación.

Externo

Especifica la lista de paquetes a omitir de la compilación. Para más información, consulta [Externo](#) en el sitio web de esbuild.

Formato

Especifica el formato de salida de los JavaScript archivos generados en la aplicación. Para más información, consulta [Formato](#) en el sitio web de esbuild.

Programa de carga

Especifica la lista de configuraciones para cargar datos para un tipo de archivo determinado.

MainFields

Especifica qué campos de `package.json` se deben intentar importar al resolver un paquete. El valor predeterminado es `main,module`.

Minify

Especifica si se debes minimizar el código de output agrupado. El valor predeterminado es `true`.

OutExtension

Personalice la extensión de los archivos que genera esbuild. Para más información, consulta [Out extension](#) en el sitio web de esbuild.

Sourcemap

Especifica si el paquete produce un archivo de mapa de origen. El valor predeterminado es `false`.

Cuando se establece en `true`, `NODE_OPTIONS: --enable-source-maps` se añade a las variables de entorno de la función de Lambda y se genera un mapa de origen que se incluye en la función.

Como alternativa, cuando `NODE_OPTIONS: --enable-source-maps` se incluye en las variables de entorno de la función, `Sourcemap` se establece automáticamente en `true`.

En caso de conflicto, `Sourcemap: false` tiene prioridad sobre `NODE_OPTIONS: --enable-source-maps`.

Note

De forma predeterminada, Lambda cifra todas las variables de entorno en reposo con AWS Key Management Service (AWS KMS). Al utilizar mapas de origen, para que la implementación se realice correctamente, tu rol de ejecución de la función debe tener permiso para realizar la acción de `kms:Encrypt`.

SourcesContent

Especifica si se debes incluir el código fuente en el archivo de mapa fuente. Configure esta propiedad cuando Sourcemap esté establecido en 'true'.

- Especifique `SourcesContent: 'true'` para incluir todo el código fuente.
- Especifique `SourcesContent: 'false'` para excluir todo el código fuente. Esto se traduce en tamaños más pequeños de los archivos de los mapas fuente, lo que resulta útil en la producción al reducir los tiempos de puesta en marcha. Sin embargo, el código fuente no estará disponible en el depurador.

El valor predeterminado es `SourcesContent: true`.

Para más información, consulta [Contenido de las fuentes](#) en el sitio web de esbuild.

Destinatario

Especifica la versión de destino para ECMAScript. El valor predeterminado es `es2020`.

TypeScript Ejemplo de función de evento Lambda

El siguiente fragmento de AWS SAM plantilla de ejemplo usa esbuild para crear una función Node.js Lambda a partir del código. TypeScript `hello-world/app.ts`

```
Resources:
  HelloWorldFunction:
    Type: AWS::Serverless::Function
    Properties:
      CodeUri: hello-world/
      Handler: app.handler
      Runtime: nodejs20.x
      Architectures:
        - x86_64
      Events:
        HelloWorld:
          Type: Api
          Properties:
            Path: /hello
            Method: get
      Environment:
        Variables:
          NODE_OPTIONS: --enable-source-maps
```

```
Metadata:
  BuildMethod: esbuild
  BuildProperties:
    Format: esm
    Minify: false
    OutExtension:
      - .js=.mjs
    Target: "es2020"
    Sourcemap: true
    EntryPoints:
      - app.ts
    External:
      - "<package-to-exclude>"
```

Creación de funciones de eventos Lambda de .NET con compilación AOT nativa en AWS SAM

Cree y empaquete sus funciones de AWS Lambda eventos de .NET 8 con AWS Serverless Application Model (AWS SAM), utilizando la compilación nativa Ahead-of-Time (AOT) para mejorar los tiempos de arranque en AWS Lambda frío.

Temas

- [Descripción general de AOT nativo para .NET 8](#)
- [Utilización AWS SAM con sus funciones de eventos Lambda de .NET 8](#)
- [Requisitos previos de instalación](#)
- [Defina las funciones de eventos Lambda de .NET 8 en su AWS SAM plantilla](#)
- [Crea tu aplicación con la de . AWS SAM CLI](#)
- [Más información](#)

Descripción general de AOT nativo para .NET 8

Históricamente, las funciones de Lambda .NET tienen tiempos de inicio en frío que afectan a la experiencia del usuario, la latencia del sistema y los costos de uso de las aplicaciones sin servidor. Con la compilación de AOT nativo de .NET, puede mejorar los tiempos de arranque en frío de sus funciones de Lambda. Para obtener más información sobre el AOT nativo para .NET 8, consulte [Uso del AOT nativo en el repositorio de Dotnet](#). GitHub

Utilización AWS SAM con sus funciones de eventos Lambda de .NET 8

Siga estos pasos para configurar las funciones de Lambda de .NET 8 con AWS Serverless Application Model (AWS SAM):

- Instala los requisitos previos en su máquina de desarrollo.
- Defina las funciones Lambda de .NET 8 en su plantilla. AWS SAM
- Cree su aplicación con. AWS SAMCLI

Requisitos previos de instalación

A continuación, se indican los requisitos previos.

- El AWS SAMCLI
- La CLI de .NET Core
- La herramienta global Amazon.Lambda.Tools de .NET Core
- Docker

Instala la AWS SAM CLI

1. Para comprobar si ya tienes la CLI de AWS SAM instalada, ejecuta lo siguiente:

```
sam --version
```

2. Para instalar el AWS SAMCLI, consulte [Instala la AWS SAM CLI](#).
3. Para actualizar una versión instalada del AWS SAMCLI, consulte [Actualización de AWS SAM CLI](#).

Instalar la CLI de .NET Core

1. Para descargar e instalar la CLI de .NET Core, consulta [Descargar .NET](#) desde el sitio web de Microsoft.
2. Para obtener más información sobre la CLI de .NET Core, consulta la [CLI de .NET Core](#) en la Guía para desarrolladores de AWS Lambda .

Instale la herramienta global Amazon.Lambda.Tools de .NET Core

1. Ejecuta el siguiente comando:

```
dotnet tool install -g Amazon.Lambda.Tools
```

2. Si ya tienes la herramienta instalada, puedes asegurarse de que es la versión más reciente usando el siguiente comando:

```
dotnet tool update -g Amazon.Lambda.Tools
```

3. Para obtener más información acerca de la herramienta global Amazon.Lambda.Tools .NET Core, consulte el repositorio [CLI de AWS extensiones para .NET](#) en GitHub.

Instalación Docker

- Para compilar con AOT nativo, es necesario instalar Docker. Para obtener las instrucciones de instalación, consulta [Instalación de Docker para usarlo con la de AWS SAM CLI](#).

Defina las funciones de eventos Lambda de .NET 8 en su AWS SAM plantilla

Para definir una función Lambda de .NET8 en la AWS SAM plantilla, haga lo siguiente:

1. Ejecute el siguiente comando desde el directorio de inicio que quiera:

```
sam init
```

2. Seleccione AWS Quick Start Templates para elegir una plantilla de inicio.
3. Elige la plantilla Hello World Example.
4. Introduzca n para no utilizar la versión ejecutable y el tipo de paquete más populares.
5. Para la versión ejecutable, elija dotnet8.
6. Para el tipo de paquete, elija Zip.
7. Para su plantilla de inicio, elija Hello World Example using native AOT.

Instalación Docker

- Para compilar con AOT nativo, es necesario instalar Docker. Para obtener las instrucciones de instalación, consulta [Instalación de Docker para usarlo con la de AWS SAM CLI](#).

```
Resources:
HelloWorldFunction:
  Type: AWS::Serverless::Function
  Properties:
    CodeUri: ./src/HelloWorldAot/
    Handler: bootstrap
    Runtime: dotnet8
    Architectures:
      - x86_64
    Events:
      HelloWorldAot:
        Type: Api
        Properties:
          Path: /hello
          Method: get
```

Note

Cuando la Event propiedad de an `AWS::Serverless::Function` se establece en `Api`, pero no se especifica, AWS SAM genera el recurso. `RestApiId` `AWS::ApiGateway::RestApi` CloudFormation

Crea tu aplicación con la de . AWS SAM CLI

Desde el directorio raíz del proyecto, ejecute el comando `sam build` para empezar a crear la aplicación. Si la propiedad `PublishAot` se ha definido en el archivo de proyecto de .NET 8, la AWS SAM de CLI se compilará con la compilación de AOT nativo. Para obtener más información sobre la propiedad `PublishAot`, consulta [Implementación nativa de AOT](#) en la documentación de .NET de Microsoft.

Para crear su función, AWS SAMCLI invoca la CLI de .NET Core, que utiliza la herramienta global `Amazon.Lambda.Tools` de .NET Core.

Note

Al compilar, si existe un archivo `.sln` en el mismo directorio o en el directorio principal del proyecto, el directorio que contiene el archivo `.sln` se montará en el contenedor. Si no se encuentra ningún archivo `.sln`, solo se montará la carpeta del proyecto. Por lo tanto, si

está creando una aplicación para varios proyectos, asegúrese de que el archivo `.sln` esté ubicado en la propiedad.

Más información

Para obtener más información sobre la creación de funciones de Lambda de .NET 8, consulte [Introducing the .NET 8 runtime for AWS Lambda](#).

Para obtener más información sobre el comando `sam build`, consulta [compilación de sam](#).

Creación de funciones de eventos de Rust Lambda con Cargo Lambda in AWS SAM

Utilice la interfaz de línea de AWS Serverless Application Model comandos (AWS SAMCLI) con sus funciones de AWS Lambda eventos de Rust.

Temas

- [Requisitos previos](#)
- [Configuración AWS SAM para usar con las funciones de eventos de Rust Lambda](#)
- [Ejemplos](#)
- [La optimización de Rust se integra GitHub Acciones](#)

Requisitos previos

Lenguaje Rust

Para instalar Rust, consulta [Instalar Rust](#) en el sitio web del lenguaje Rust.

Cargo Lambda

La CLI de AWS SAM requiere la instalación de [Cargo Lambda](#), un subcomando para Cargo. Para obtener instrucciones sobre la instalación, consulta [Instalación](#) en la Documentación de Cargo Lambda.

Docker

Para crear y probar funciones de Lambda Rust es necesario disponer de Docker. Para obtener las instrucciones de instalación, consulta [Instalar Docker](#).

Configuración AWS SAM para usar con las funciones de eventos de Rust Lambda

Paso 1: Configura tu AWS SAM plantilla

Configure su AWS SAM plantilla con lo siguiente:

- **Binario:** opcional. Especifique cuándo un solo Cargo paquete define más de un binario para identificar qué binario crear para esta función. No necesita esta propiedad cuando cada función es su propio Cargo paquete, como en un Cargo espacio de trabajo.
- **BuildMethod** – `rust-cargolambda`.
- **CodeUri**— ruta a su `Cargo.toml` archivo.
- **Controlador:** `bootstrap`.
- **Tiempo de ejecución:** `provided.al2023`.

Para obtener más información sobre los tiempos de ejecución personalizados, consulta los tiempos de [AWS Lambda ejecución personalizados](#) en la Guía AWS Lambda para desarrolladores.

Este es un ejemplo de una plantilla configurada AWS SAM :

```
AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31
...
Resources:
  MyFunction:
    Type: AWS::Serverless::Function
    Metadata:
      BuildMethod: rust-cargolambda
      BuildProperties: function_a
    Properties:
      CodeUri: ./rust_app
      Handler: bootstrap
      Runtime: provided.al2023
  ...
```

Paso 2: utilice la AWS SAM CLI con su función de eventos de Rust Lambda

Usa cualquier AWS SAMCLI comando con tu AWS SAM plantilla. Para obtener más información, consulte [AWS SAM CLI](#).

Ejemplos

Ejemplo de Hello World

En este ejemplo, creamos la aplicación de muestra Hello World usando Rust como nuestro tiempo de ejecución.

En primer lugar, inicializamos una nueva aplicación sin servidor utilizando `sam init`. Durante el flujo interactivo, seleccionamos la aplicación Hello World y elegimos el tiempo de ejecución de Rust.

```
$ sam init
...
Which template source would you like to use?
    1 - AWS Quick Start Templates
    2 - Custom Template Location
Choice: 1

Choose an AWS Quick Start application template
    1 - Hello World Example
    2 - Multi-step workflow
    3 - Serverless API
    ...
Template: 1

Use the most popular runtime and package type? (Python and zip) [y/N]: ENTER

Which runtime would you like to use?
    1 - dotnet8
    2 - dotnet6
    3 - go (provided.al2)
    ...
    18 - python3.11
    19 - python3.10
    20 - ruby4.0
    21 - ruby3.3
    22 - ruby3.2
    23 - rust (provided.al2)
    24 - rust (provided.al2023)
Runtime: 24

Based on your selections, the only Package type available is Zip.
We will proceed to selecting the Package type as Zip.

Based on your selections, the only dependency manager available is cargo.
```

We will proceed copying the template using cargo.

Would you like to enable X-Ray tracing on the function(s) in your application? [y/N]: *ENTER*

Would you like to enable monitoring using CloudWatch Application Insights?
For more info, please view <https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/cloudwatch-application-insights.html> [y/N]: *ENTER*

Project name [sam-app]: *hello-rust*

```
-----
Generating application:
-----
```

```
Name: hello-rust
Runtime: rust (provided.al2023)
Architectures: x86_64
Dependency Manager: cargo
Application Template: hello-world
Output Directory: .
Configuration file: hello-rust/samconfig.toml
```

Next steps can be found in the README file at hello-rust/README.md

Commands you can use next

```
=====
[*] Create pipeline: cd hello-rust && sam pipeline init --bootstrap
[*] Validate SAM template: cd hello-rust && sam validate
[*] Test Function in the Cloud: cd hello-rust && sam sync --stack-name {stack-name} --
watch
```

La siguiente es la estructura de nuestra aplicación Hello World:

```
hello-rust
### README.md
### events
#   ### event.json
### rust_app
#   ### Cargo.toml
#   ### src
#       ### main.rs
### samconfig.toml
```

```
### template.yaml
```

En nuestra AWS SAM plantilla, nuestra Rust función se define de la siguiente manera:

```
AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31
...
Resources:
  HelloWorldFunction:
    Type: AWS::Serverless::Function
    Metadata:
      BuildMethod: rust-cargolambda
    Properties:
      CodeUri: ./rust_app
      Handler: bootstrap
      Runtime: provided.al2023
      Architectures:
        - x86_64
      Events:
        HelloWorld:
          Type: Api
          Properties:
            Path: /hello
            Method: get
```

A continuación, ejecuta `sam build` para crear e implementar la aplicación. La CLI de AWS SAM crea un directorio de `.aws-sam` y organiza allí nuestros artefactos de construcción. Nuestra función se crea utilizando Cargo Lambda y se almacena como un archivo binario ejecutable en `.aws-sam/build/HelloWorldFunction/bootstrap`.

Note

Si planea ejecutar el comando `sam local invoke` en macOS, tiene que crear funciones diferentes antes de la invocación. Para ello, utilice el siguiente comando:

- `SAM_BUILD_MODE=debug sam build`

Este comando solo es necesario si se van a realizar pruebas locales. No se recomienda al compilar para la implementación.

```

hello-rust$ sam build
Starting Build use cache
Cache is invalid, running build and copying resources for following functions
(HelloWorldFunction)
Building codeuri: /Users/.../hello-rust/rust_app runtime: provided.al2023 metadata:
{'BuildMethod': 'rust-cargolambda'} architecture: x86_64 functions: HelloWorldFunction
Running RustCargoLambdaBuilder:CargoLambdaBuild
Running RustCargoLambdaBuilder:RustCopyAndRename

Build Succeeded

Built Artifacts   : .aws-sam/build
Built Template    : .aws-sam/build/template.yaml

Commands you can use next
=====
[*] Validate SAM template: sam validate
[*] Invoke Function: sam local invoke
[*] Test Function in the Cloud: sam sync --stack-name {{stack-name}} --watch
[*] Deploy: sam deploy --guided

```

A continuación, implementamos nuestra aplicación usando `sam deploy --guided`.

```

hello-rust$ sam deploy --guided

Configuring SAM deploy
=====

Looking for config file [samconfig.toml] : Found
Reading default arguments : Success

Setting default arguments for 'sam deploy'
=====
Stack Name [hello-rust]: ENTER
AWS Region [us-west-2]: ENTER
#Shows you resources changes to be deployed and require a 'Y' to initiate
deploy
Confirm changes before deploy [Y/n]: ENTER
#SAM needs permission to be able to create roles to connect to the resources in
your template
Allow SAM CLI IAM role creation [Y/n]: ENTER
#Preserves the state of previously provisioned resources when an operation
fails

```


Disable rollback [y/N]: **ENTER**

HelloWorldFunction may not have authorization defined, Is this okay? [y/N]: **y**

Save arguments to configuration file [Y/n]: **ENTER**

SAM configuration file [samconfig.toml]: **ENTER**

SAM configuration environment [default]: **ENTER**

Looking for resources needed for deployment:

...

Uploading to hello-rust/56ba6585d80577dd82a7eaaee5945c0b 817973 / 817973
(100.00%)

Deploying with following values

=====

Stack name	: hello-rust
Region	: us-west-2
Confirm changeset	: True
Disable rollback	: False
Deployment s3 bucket	: aws-sam-cli-managed-default-samclisam-s3-demo-
bucket-1a4x26zbcdkqr	
Capabilities	: ["CAPABILITY_IAM"]
Parameter overrides	: {}
Signing Profiles	: {}

Initiating deployment

=====

Uploading to hello-rust/a4fc54cb6ab75dd0129e4cdb564b5e89.template 1239 / 1239
(100.00%)

Waiting for changeset to be created..

CloudFormation stack changeset

Operation	LogicalResourceId	ResourceType	
Replacement			
+ Add	HelloWorldFunctionHelloW	AWS::Lambda::Permission	N/A
	orldPermissionProd		

...

```
-----
Changeset created successfully. arn:aws:cloudformation:us-
west-2:012345678910:changeSet/samcli-deploy1681427201/
f0ef1563-5ab6-4b07-9361-864ca3de6ad6
```

```
Previewing CloudFormation changeset before deployment
=====
```

```
Deploy this changeset? [y/N]: y
```

```
2023-04-13 13:07:17 - Waiting for stack create/update to complete
```

```
CloudFormation events from stack operations (refresh every 5.0 seconds)
```

```
-----
ResourceStatus      ResourceType      LogicalResourceId
ResourceStatusReason
-----
CREATE_IN_PROGRESS  AWS::IAM::Role    HelloWorldFunctionRole  -
CREATE_IN_PROGRESS  AWS::IAM::Role    HelloWorldFunctionRole
Resource creation
...
-----
```

```
CloudFormation outputs from deployed stack
```

```
-----
Outputs
```

```
-----
Key                  HelloWorldFunctionIamRole

Description          Implicit IAM Role created for Hello World function

Value               arn:aws:iam::012345678910:role/hello-rust-
HelloWorldFunctionRole-10II2P13AUDUY

Key                  HelloWorldApi

Description          API Gateway endpoint URL for Prod stage for Hello World function

Value               https://ggdxec9le9.execute-api.us-west-2.amazonaws.com/Prod/hello/
-----
```

Key	HelloWorldFunction
Description	Hello World Lambda Function ARN
Value	arn:aws:lambda:us-west-2:012345678910:function:hello-rust-HelloWorldFunction-yk4HzGzYeZBj

Successfully created/updated stack - hello-rust in us-west-2

Para realizar las pruebas, podemos invocar nuestra función de Lambda mediante el punto de conexión de la API.

```
$ curl https://ggdxec91e9.execute-api.us-west-2.amazonaws.com/Prod/hello/Hello World!%
```

Para probar nuestra función localmente, primero nos aseguramos de que la propiedad Architectures de nuestra función coincida con nuestro equipo local.

```
...
Resources:
  HelloWorldFunction:
    Type: AWS::Serverless::Function # More info about Function Resource:
    https://github.com/aws-labs/serverless-application-model/blob/master/versions/2016-10-31.md#awsserverlessfunction
    Metadata:
      BuildMethod: rust-cargolambda # More info about Cargo Lambda: https://github.com/cargo-lambda/cargo-lambda
    Properties:
      CodeUri: ./rust_app # Points to dir of Cargo.toml
      Handler: bootstrap # Do not change, as this is the default executable name
                        produced by Cargo Lambda
      Runtime: provided.al2023
      Architectures:
        - arm64
...
```

Como ya modificamos nuestra arquitectura de x86_64 a arm64 en este ejemplo, ejecutamos `sam build` para actualizar nuestros artefactos de construcción. A continuación, ejecutamos `sam local invoke` para invocar nuestra función de forma local.

```
hello-rust$ sam local invoke
Invoking bootstrap (provided.al2023)
Local image was not found.
Removing rapid images for repo public.ecr.aws/sam/emulation-provided.al2023
Building
  image.....
Using local image: public.ecr.aws/lambda/provided:al2023-rapid-arm64.

Mounting /Users/.../hello-rust/.aws-sam/build/HelloWorldFunction as /var/
task:ro,delegated, inside runtime container
START RequestId: fbc55e6e-0068-45f9-9f01-8e2276597fc6 Version: $LATEST
{"statusCode":200,"body":"Hello World!"}END RequestId:
fbc55e6e-0068-45f9-9f01-8e2276597fc6
REPORT RequestId: fbc55e6e-0068-45f9-9f01-8e2276597fc6  Init Duration: 0.68 ms
Duration: 130.63 ms      Billed Duration: 131 ms      Memory Size: 128 MB      Max Memory
Used: 128 MB
```

Proyecto de función de evento Lambda única

Este es un ejemplo de una aplicación sin servidor que contiene una función Rust de Lambda.

Estructura del directorio del proyecto:

```
.
### Cargo.lock
### Cargo.toml
### src
#   ### main.rs
### template.yaml
```

AWS SAM plantilla:

```
AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31
...
Resources:
  MyFunction:
```

```
Type: AWS::Serverless::Function
Metadata:
  BuildMethod: rust-cargolambda
Properties:
  CodeUri: ./
  Handler: bootstrap
  Runtime: provided.al2023
...
```

Proyecto de función de eventos Lambda múltiple

Este es un ejemplo de una aplicación sin servidor que contiene varias funciones de Rust Lambda, organizadas como Cargo un espacio de trabajo.

Recomendamos un Cargo espacio de trabajo para aplicaciones con múltiples funciones de Rust Lambda. Cada función es su propio paquete, por lo que las funciones pueden declarar dependencias independientes y compartir código común a través de un paquete de biblioteca. Cada paquete produce un único binario con el nombre del paquete, por lo que no es necesario establecer la propiedad de Binary compilación.

Estructura del directorio del proyecto:

```
.
### Cargo.lock
### Cargo.toml
### function_a
#   ### Cargo.toml
#   ### src
#       ### main.rs
### function_b
#   ### Cargo.toml
#   ### src
#       ### main.rs
### template.yaml
```

`Cargo.toml` Archivo de espacio de trabajo, en la raíz del proyecto:

```
[workspace]
resolver = "2"
members = [
  "function_a",
```

```

    "function_b",
  ]

[workspace.dependencies]
lambda_runtime = "0.13"
serde = { version = "1", features = ["derive"] }
tokio = { version = "1", features = ["macros", "rt"] }

```

Cargo.toml archivo para cada función, como `function_a/Cargo.toml`:

```

[package]
name = "function_a"
version = "0.1.0"
edition = "2021"

[dependencies]
lambda_runtime = { workspace = true }
serde = { workspace = true }
tokio = { workspace = true }

```

AWS SAM plantilla. La `CodeUri` de cada función apunta al directorio de paquetes de esa función:

```

AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31
...
Resources:
  FunctionA:
    Type: AWS::Serverless::Function
    Metadata:
      BuildMethod: rust-cargolambda
    Properties:
      CodeUri: ./function_a
      Handler: bootstrap
      Runtime: provided.al2023
  FunctionB:
    Type: AWS::Serverless::Function
    Metadata:
      BuildMethod: rust-cargolambda
    Properties:
      CodeUri: ./function_b
      Handler: bootstrap
      Runtime: provided.al2023

```

 Note

AWS SAMCLI crea todas las funciones del espacio de trabajo en el `target` directorio compartido del espacio de trabajo, por lo que Cargo compila las dependencias compartidas una vez en lugar de una para cada función. Este comportamiento requiere la AWS SAMCLI versión 1.165.0 o posterior. En las versiones anteriores, cada función se crea en su propio `target` directorio y se vuelve a compilar el árbol de dependencias completo para cada función, lo que hace que las compilaciones sean más lentas a medida que se agregan funciones.

Asigne a cada paquete de funciones un nombre binario único. Los nombres de los paquetes son únicos dentro de un espacio de trabajo, por lo que el nombre binario predeterminado ya es único. Si reemplaza el nombre binario por una `[[bin]]` sección, no dé a dos paquetes el mismo nombre binario. Se compilan en la misma ruta del `target` directorio compartido y se sobrescriben entre sí. AWS SAMCLI registra una advertencia cuando lo detecta.

Como alternativa, un solo paquete puede definir varios binarios. En ese caso, utilice la propiedad `Binary build` para seleccionar el binario de cada función:

```
Resources:
  FunctionA:
    Type: AWS::Serverless::Function
    Metadata:
      BuildMethod: rust-cargolambda
      BuildProperties:
        Binary: function_a
    Properties:
      CodeUri: ./
      Handler: bootstrap
      Runtime: provided.al2023
```

La optimización de Rust se integra GitHub Acciones

Las compilaciones de Rust requieren un uso intensivo de cómputo y un ejecutor de integración continua comienza sin artefactos compilados. Las aplicaciones con varias funciones que comparten grandes dependencias, como una AWSSDK, pueden dedicar la mayor parte de su tiempo de compilación a compilar las mismas dependencias. Las siguientes prácticas reducen el tiempo de compilación. GitHub Actions

Utilice AWS SAMCLI la versión 1.165.0 o posterior para los espacios de trabajo

La versión 1.165.0 y las posteriores compilan todos los miembros de un Cargo espacio de trabajo en el `target` directorio compartido del espacio de trabajo, de modo que las dependencias compartidas se compilan una vez por compilación en lugar de una para cada función. Especifica la versión mínima cuando instales la versión AWS SAMCLI para que una compilación no recaiga silenciosamente en un comportamiento más lento.

Almacene en caché el Cargo registro y el directorio **target**

Almacene en caché el Cargo registro (`~/.cargo/registry` y `~/.cargo/git/db`) y el `target` directorio del espacio de trabajo entre ejecuciones, de modo que las dependencias no modificadas se restauren en lugar de volver a compilarse. Utilice una caché independiente para cada destino de compilación. Un trabajo que compila artefactos de publicación de forma cruzada produce artefactos diferentes a los que `arm64` produce un trabajo para el que se compila de forma nativa `x86_64`, por lo que una caché compartida nunca coincide.

Incluye la configuración de compilación en la clave de caché

Cargo incluye ajustes como `opt-level` la huella digital que utiliza para decidir si un artefacto compilado se puede reutilizar. `codegen-units` Si cambias la `[profile.release]` sección del `Cargo.toml` archivo de tu espacio de trabajo sin cambiar la clave de caché, la caché se restaura, pero todas las cajas se vuelven a compilar de todos modos. Incluye un hash del `Cargo.toml` archivo del espacio de trabajo en la clave de caché para que al cambiar la configuración de un perfil se inicie una nueva caché.

Confirme su **Cargo.lock** archivo

Las funciones Lambda son ejecutables, así que confirme su archivo. `Cargo.lock` Esto le proporciona compilaciones reproducibles y una clave de caché estable que solo cambia cuando cambian sus dependencias.

Ajuste el perfil de versión para tener en cuenta el tiempo de construcción y el arranque en frío

El código de la función se vuelve a compilar en cada ejecución, ya que cambia con más frecuencia que las dependencias. El perfil de versión predeterminado optimiza el rendimiento en tiempo de ejecución, algo que muchas funciones de Lambda no necesitan. Al optimizar el tamaño, se obtienen binarios más pequeños, lo que también contribuye al tiempo de arranque en frío. Además, si se aumenta el número de unidades de generación de código, se aumenta el paralelismo durante la compilación. Deje desactivada la optimización del tiempo de enlace (`lto`), ya que ralentiza la compilación. Añada lo siguiente a su `Cargo.toml` archivo de espacio de trabajo:


```
[profile.release]
opt-level = "s"
codegen-units = 256
lto = false
strip = true
```

Mida el efecto en su propia aplicación. Estos ajustes cambian una pequeña cantidad de rendimiento en tiempo de ejecución por tiempo de compilación y tamaño binario.

Evite duplicar las ejecuciones del flujo de trabajo

Un flujo de trabajo que se ejecuta en ambos `pull_request` eventos `push` y se ejecuta dos veces para la misma confirmación. GitHub Actions Las cachés se clasifican por rama y solicitud de extracción, por lo que las dos ejecuciones escriben en ámbitos de caché diferentes y ninguna reutiliza la caché de la otra. Usa un grupo de simultaneidad que esté escrito en la confirmación principal, de modo que solo una ejecución genere cada confirmación.

El siguiente flujo de trabajo crea un Cargo espacio de trabajo de funciones de Rust Lambda para `arm64` y aplica las prácticas anteriores:

```
name: Build

on:
  push:
    branches: [main]
  pull_request:

# Collapse the push and pull_request runs for the same commit into a single run.
concurrency:
  group: ${{ github.workflow }}-${{ github.event.pull_request.head.sha || github.sha }}
  cancel-in-progress: true

jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v5

      - uses: dtolnay/rust-toolchain@stable
        with:
          targets: aarch64-unknown-linux-gnu
```

```
# Cache the Cargo registry and the workspace target directory. The key covers
# the compilation target, Cargo.lock, and the workspace Cargo.toml, so that
# changing a dependency or a release profile setting starts a new cache
# instead of restoring one whose artifacts Cargo discards.
- uses: actions/cache@v4
  with:
    path: |
      ~/.cargo/registry/index
      ~/.cargo/registry/cache
      ~/.cargo/git/db
      target
    key: cargo-arm64-${{ hashFiles('Cargo.lock', 'Cargo.toml') }}
    restore-keys: |
      cargo-arm64-

- name: Install build tools
  run: pip install cargo-lambda 'aws-sam-cli>=1.165.0'

- name: Build
  run: sam build
```

La `restore-keys` entrada permite iniciar una ejecución desde la caché más reciente cuando la clave no coincide exactamente, de modo que un cambio de dependencia reutiliza las cajas que no se han modificado en lugar de volver a compilarlo todo.

Creación de funciones de eventos Lambda de Python con uv in AWS SAM

Esta función se encuentra en una versión preliminar AWS SAM y está sujeta a cambios.

Utilice la interfaz de línea de AWS Serverless Application Model comandos (AWS SAMCLI) con uv un rápido instalador y solucionador de paquetes de Python para crear sus funciones de AWS Lambda eventos de Python.

Temas

- [Requisitos previos](#)
- [Configuración AWS SAM para usar con las funciones de eventos Lambda de Python y uv](#)
- [Ejemplos](#)

Requisitos previos

Python

Para instalar Python, consulta [Descargar Python](#) en el sitio web de Python.

uv

AWS SAM CLI Requiere la instalación de [uv](#) un instalador y solucionador de paquetes de Python extremadamente rápido. Para obtener instrucciones sobre la instalación, consulta [Instalación](#) en la Documentación de uv.

Acceda a la característica beta de la CLI de AWS SAM.

Como esta característica está en versión preliminar, debes acceder usando uno de los siguientes métodos:

1. Usa las variables de entorno: `SAM_CLI_BETA_PYTHON_UV=1`.
2. Añade lo siguiente a tu archivo `samconfig.toml`:

```
[default.build.parameters]
beta_features = true
[default.sync.parameters]
beta_features = true
```

3. Utiliza la opción `--beta-features` cuando utiliza un comando de la CLI de AWS SAM compatible. Por ejemplo:

```
$ sam build --beta-features
```

4. Elija la opción y cuando la CLI de AWS SAM le pida que acceda. A continuación, se muestra un ejemplo:

```
$ sam build
Starting Build use cache
Build method "python-uv" is a beta feature.
Please confirm if you would like to proceed
You can also enable this beta feature with "sam build --beta-features". [y/N]: y
```

Configuración AWS SAM para usar con las funciones de eventos Lambda de Python y uv

Paso 1: Configura tu AWS SAM plantilla

Configura tu AWS SAM plantilla con lo siguiente:

- BuildMethod – python-uv.
- CodeUri— ruta al directorio de códigos de funciones que contiene `pyproject.toml` o `requirements.txt`.
- Controlador: su controlador de funciones (por ejemplo, `app.lambda_handler`).
- Runtime: versión en tiempo de ejecución de Python (p. ej., `python3.12`).

Este es un ejemplo de una AWS SAM plantilla configurada:

```
AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31
...
Resources:
  MyFunction:
    Type: AWS::Serverless::Function
    Properties:
      CodeUri: ./my_function
      Handler: app.lambda_handler
      Runtime: python3.12
    Metadata:
      BuildMethod: python-uv
  ...
```

Ejemplos

Ejemplo de Hello World

En este ejemplo, creamos un ejemplo de aplicación Hello World utilizando Python uv como gestor de paquetes.

uv puede usar `pyproject.toml` o `requirements.txt` leer las dependencias. Si se proporcionan ambos, sam build leerá desde `requirements.txt` las dependencias.

La siguiente es la estructura de nuestra aplicación Hello World:

```
hello-python-uv
### README.md
### events
#   ### event.json
### hello_world
#   ### __init__.py
#   ### app.py
#   ### pyproject.toml
### samconfig.toml
### template.yaml
```

Archivo `pyproject.toml`:

```
[project]
name = "my-function"
version = "0.1.0"
requires-python = ">=3.12"
dependencies = [
    "requests>=2.31.0",
    "boto3>=1.28.0",
]
```

En nuestra AWS SAM plantilla, nuestra función de Python se define de la siguiente manera:

```
AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31
...
Resources:
  HelloWorldFunction:
    Type: AWS::Serverless::Function
    Properties:
      CodeUri: hello_world/
      Handler: app.lambda_handler
      Runtime: python3.12
      Architectures:
        - x86_64
    Metadata:
      BuildMethod: python-uv
```

A continuación, ejecuta `sam build` para crear e implementar la aplicación. La CLI de AWS SAM crea un directorio de `.aws-sam` y organiza allí nuestros artefactos de construcción.

Las dependencias de nuestras funciones se instalan uv y almacenan en `.aws-sam/build/HelloWorldFunction/`.

```
hello-python-uv$ sam build
Starting Build use cache
Build method "python-uv" is a beta feature.
Please confirm if you would like to proceed
You can also enable this beta feature with "sam build --beta-features". [y/N]: y

Experimental features are enabled for this session.
Visit the docs page to learn more about the AWS Beta terms https://aws.amazon.com/
service-terms/.

Cache is invalid, running build and copying resources for following functions
(HelloWorldFunction)
Building codeuri: /Users/.../hello-python-uv/hello_world runtime: python3.12 metadata:
{'BuildMethod': 'python-uv'} architecture: x86_64 functions: HelloWorldFunction
Running PythonUvBuilder:UvBuild
Running PythonUvBuilder:CopySource

Build Succeeded

Built Artifacts   : .aws-sam/build
Built Template    : .aws-sam/build/template.yaml

Commands you can use next
=====
[*] Validate SAM template: sam validate
[*] Invoke Function: sam local invoke
[*] Test Function in the Cloud: sam sync --stack-name {{stack-name}} --watch
[*] Deploy: sam deploy --guided
```

Note

El método de python-uv compilación se configura por función en la Metadata sección. Cada función de la plantilla puede utilizar un método de creación diferente, lo que te permite mezclar funciones uv basadas con funciones pip basadas en la misma AWS SAM plantilla. Si no se especifica ningún método de compilación, pip se utiliza de forma predeterminada.

Creación de funciones de eventos de Lambda con tiempos de ejecución personalizados en AWS SAM

Puede usar el [compilación de sam](#) comando para crear los tiempos de ejecución personalizados necesarios para la función de eventos de Lambda. Para declarar que la función de Lambda utilizarás un tiempo de ejecución personalizado, especifique `Runtime: provided` para la función.

Para crear un tiempo de ejecución personalizado, declare el atributo de recurso Metadata con una entrada `BuildMethod: makefile`. Proporcione el archivo Make personalizado, en el que se declara el destino de compilación del formulario `build-function-logical-id` que contiene los comandos de compilación para su tiempo de ejecución. El archivo Make se encarga de compilar el tiempo de ejecución personalizado, si es necesario, y de copiar los artefactos de compilación en la ubicación adecuada para los pasos posteriores del flujo de trabajo. La ubicación del archivo Make viene especificada por la propiedad `CodeUri` del recurso de la función y debes llamarse `Makefile`.

Ejemplos

Ejemplo 1: tiempo de ejecución personalizado para una función escrita en Rust

Note

Recomendamos crear funciones de Lambda con Cargo Lambda. Para obtener más información, consulte [Creación de funciones de eventos de Rust Lambda con Cargo Lambda in AWS SAM](#).

La siguiente AWS SAM plantilla declara una función que usa un tiempo de ejecución personalizado para una función Lambda escrita en Rust e indica `sam build` que se ejecuten los comandos para el `build-HelloRustFunction` destino de compilación.

```
Resources:
  HelloRustFunction:
    Type: AWS::Serverless::Function
    Properties:
      FunctionName: HelloRust
      Handler: bootstrap.is.real.handler
      Runtime: provided
      MemorySize: 512
      CodeUri: .
    Metadata:
```

```
BuildMethod: makefile
```

El archivo Make siguiente contiene el destino de compilación y los comandos que se ejecutarán. Tenga en cuenta que la propiedad `CodeUri` está establecida en `.`, por lo que el archivo Make debes estar ubicado en el directorio raíz del proyecto (es decir, el mismo directorio que el archivo de plantilla de la aplicación de AWS SAM). El nombre del archivo debes ser `Makefile`.

```
build-HelloRustFunction:
  cargo build --release --target x86_64-unknown-linux-musl
  cp ./target/x86_64-unknown-linux-musl/release/bootstrap $(ARTIFACTS_DIR)
```

Para obtener más información sobre cómo configurar el entorno de desarrollo para ejecutar el comando `cargo build` en el anterior `makefile`, consulta la publicación de blog [Tiempo de ejecución de Rust para AWS Lambda](#).

Ejemplo 2: generador de archivos Makefile para Python3.12 (alternativa a usar el generador incluido)

Es posible que quieras utilizar una biblioteca o un módulo que no esté incluido en un paquete de creación. En este ejemplo, se muestra una AWS SAM plantilla para un entorno de Python3.12 ejecución con un generador de archivos Makefile.

```
Resources:
  HelloWorldFunction:
    Type: AWS::Serverless::Function
    Properties:
      CodeUri: hello_world/
      Handler: app.lambda_handler
      Runtime: python3.12
    Metadata:
      BuildMethod: makefile
```

El archivo Make siguiente contiene el destino de compilación y los comandos que se ejecutarán. Ten en cuenta que la propiedad `CodeUri` está establecida en `hello_world`, por lo que el archivo Make debe estar ubicado en la raíz del subdirectorio `hello_world` y el nombre del archivo debe ser `Makefile`.

```
build-HelloWorldFunction:
  cp *.py $(ARTIFACTS_DIR)
  cp requirements.txt $(ARTIFACTS_DIR)
  python -m pip install -r requirements.txt -t $(ARTIFACTS_DIR)
```



```
rm -rf $(ARTIFACTS_DIR)/bin
```

Creación de capas Lambda en AWS SAM

Se puede utilizar AWS SAM para crear capas Lambda personalizadas. Las capas Lambda permiten extraer código de una función de eventos de Lambda que luego se puede reutilizar en varias funciones de Lambda. Compilar solo capas de Lambda (en lugar de compilar toda la aplicación) puede beneficiarle de varias maneras. Puede ayudarle a reducir el tamaño de sus paquetes de implementación, separar la lógica de las funciones principales de las dependencias y permitirle compartir las dependencias entre varias funciones. Para obtener más información acerca de las capas, consulta [Capas Lambda de AWS](#), en la Guía para desarrolladores de AWS Lambda .

Cómo crear una capa Lambda en AWS SAM

Note

Antes de poder crear una capa Lambda, primero debe escribir una capa Lambda en la plantilla. AWS SAM Para obtener más información y ejemplos, consulte [Aumente la eficiencia con las capas Lambda con AWS SAM](#).

Para crear una capa personalizada, declárala en tu archivo de plantilla AWS Serverless Application Model (AWS SAM) e incluye una sección de atributos del Metadata recurso con una BuildMethod entrada. Los valores válidos BuildMethod son los identificadores de un tiempo [de ejecución de AWS Lambda](#), omakefile. Incluya una entrada BuildArchitecture para especificar las arquitecturas del conjunto de instrucciones que admite la capa. Los valores válidos para BuildArchitecture son las [arquitecturas de conjuntos de instrucciones Lambda](#).

Si quieres especificar makefile, proporciona el archivo Make personalizado, en el que se declara el destino de compilación del formulario build-*layer-logical-id* que contiene los comandos de creación de la capa. El archivo Make se encarga de compilar la capa, si es necesario, y de copiar los artefactos de compilación en la ubicación adecuada para los siguientes pasos del flujo de trabajo. La ubicación del archivo Make viene especificada por la propiedad ContentUri del recurso de capa y debe tener un nombre Makefile.

 Note

Al crear una capa personalizada, AWS Lambda depende de las variables de entorno para encontrar el código de la capa. Los tiempos de ejecución de Lambda incluyen rutas en el directorio `/opt` en el que se copia el código de capa. La estructura de carpetas de artefactos de compilación del proyecto debe coincidir con la estructura de carpetas esperada para el tiempo de ejecución para poder encontrar el código de capa personalizado.

Por ejemplo, para Python, puedes colocar el código en el subdirectorio `python/`.

Por ejemplo, para NodeJS, puedes colocar el código en el subdirectorio `nodejs/node_modules/`.

Para obtener más información, consulta [Incluir las dependencias de biblioteca en una capa](#) en la Guía para desarrolladores de AWS Lambda .

A continuación, se muestra un ejemplo de una sección de atributos de recursos Metadata.

```
Metadata:
  BuildMethod: python3.12
  BuildArchitecture: arm64
```

 Note

Si no incluye la sección de atributos del Metadata recurso, AWS SAM no crea la capa. En su lugar, copia los artefactos de compilación de la ubicación especificada en la propiedad `CodeUri` del recurso de la capa. Para obtener más información, consulta la propiedad [ContentUri](#) del tipo de recurso `AWS::Serverless::LayerVersion`.

Al incluir la sección de atributos de Metadata recursos, puede usar el [compilación de sam](#) comando para crear la capa, como un objeto independiente o como una dependencia de una AWS Lambda función.

- Como objeto independiente. Es posible que solo quieras crear el objeto de la capa, por ejemplo, cuando estés probando localmente un cambio de código en la capa y no necesites compilar toda la aplicación. Para crear la capa de forma independiente, especifica el recurso de la capa con el comando `sam build layer-logical-id`.

- Como dependencia de una función de Lambda. Al incluir el ID lógico de una capa en la propiedad `Layers` de una función de Lambda del mismo archivo de plantilla de AWS SAM, la capa es una dependencia de esa función de Lambda. Cuando esa capa también incluye una sección de atributos de recursos `Metadata` con una entrada `BuildMethod`, se crea la capa creando toda la aplicación con el comando `sam build` o especificando el recurso de la función con el comando `sam build function-logical-id`.

Ejemplos

Ejemplo de plantilla 1: Crear una capa en el entorno de ejecución de Python 3.12

La siguiente AWS SAM plantilla de ejemplo crea una capa en el entorno de ejecución de Python 3.12.

```
Resources:
  MyLayer:
    Type: AWS::Serverless::LayerVersion
    Properties:
      ContentUri: my_layer
      CompatibleRuntimes:
        - python3.12
    Metadata:
      BuildMethod: python3.12  # Required to have AWS SAM build this layer
```

Ejemplo de plantilla 2: Crear una capa con un archivo Make personalizado

En la siguiente AWS SAM plantilla de ejemplo, se utiliza una plantilla personalizada `makefile` para crear la capa.

```
Resources:
  MyLayer:
    Type: AWS::Serverless::LayerVersion
    Properties:
      ContentUri: my_layer
      CompatibleRuntimes:
        - python3.12
    Metadata:
      BuildMethod: makefile
```

`makefile` contiene el destino de compilación y los comandos que se ejecutarán. Ten en cuenta que la propiedad `ContentUri` está establecida en `my_layer`, por lo que el archivo Make debes

estar ubicado en la raíz del subdirectorio `my_layer` y el nombre del archivo debes ser `Makefile`. Tenga en cuenta también que los artefactos de construcción se copian en el `python/` subdirectorio para AWS Lambda que pueda encontrar el código de la capa.

```
build-MyLayer:
  mkdir -p "${ARTIFACTS_DIR}/python"
  cp *.py "${ARTIFACTS_DIR}/python"
  python -m pip install -r requirements.txt -t "${ARTIFACTS_DIR}/python"
```

Note

Cuando se llama a `makefile`, se activa el destino correspondiente, y los artefactos deben copiarse en la variable de entorno expuesta `$ARTIFACTS_DIR`. Para obtener más información, consulte [aws-lambda-builders](#) en GitHub.

Ejemplo de comandos de compilación de sam

Los siguientes comandos `sam build` crean capas que incluyen las secciones de atributos de recursos Metadata.

```
# Build the 'layer-logical-id' resource independently
$ sam build layer-logical-id

# Build the 'function-logical-id' resource and layers that this function depends on
$ sam build function-logical-id

# Build the entire application, including the layers that any function depends on
$ sam build
```

Pruebe su aplicación sin servidor con AWS SAM

Tras escribir y compilar la aplicación, estará preparado para probarla y comprobar que funciona correctamente. Con la interfaz de línea de AWS SAM comandos (CLI), puede probar localmente su aplicación sin servidor antes de subirla a la AWS nube. Probar la aplicación le ayuda a confirmar su funcionalidad, fiabilidad y rendimiento, al mismo tiempo que identifica los problemas (errores) que deberán abordarse.

En esta sección se proporciona orientación sobre las prácticas habituales que puede seguir para probar su aplicación. Los temas de esta sección se centran principalmente en las pruebas locales que puede realizar antes de implementarla en la AWS nube. Probar antes de la implementación le ayuda a identificar los problemas de forma proactiva, lo que reduce los costos innecesarios asociados a los problemas de implementación. Cada tema de esta sección describe una prueba que puede realizar e incluye ejemplos que muestran cómo realizarla. Tras probar la aplicación, estará preparado para depurar cualquier problema que haya encontrado.

Temas

- [Introducción a las pruebas con el sam local comando](#)
- [Invoke localmente funciones de eventos de Lambda con AWS SAM](#)
- [Ejecute API Gateway de forma local con AWS SAM](#)
- [Introducción a las pruebas en la nube con evento de prueba remoto de sam](#)
- [Introducción a las pruebas en la nube con invocación remota de sam](#)
- [Automatice las pruebas de integración local con AWS SAM](#)
- [Genere cargas útiles de eventos de muestra con AWS SAM](#)
- [Prueba y depuración de funciones duraderas](#)

Introducción a las pruebas con el sam local comando

Utilice el AWS Serverless Application Model comando Command Line Interface (AWS SAMCLI) `sam local` para probar localmente las aplicaciones sin servidor.

Para obtener una introducción al AWS SAMCLI, consulte [¿Qué es el AWS SAM CLI?](#).

Requisitos previos

Para usar `sam local`, completa lo siguiente para instalar la CLI de AWS SAM:

- [AWS SAM requisitos previos.](#)
- [Instala la AWS SAM CLI.](#)

Antes de usar `sam local`, se recomienda tener conocimientos básicos de lo siguiente:

- [Configuración de la AWS SAM de CLI.](#)
- [Crea tu aplicación en AWS SAM.](#)
- [Introducción a la construcción con AWS SAM.](#)
- [Introducción a la implementación con AWS SAM.](#)

Uso de sam local comando

Usa el comando `sam local` con cualquiera de sus subcomandos para realizar diferentes tipos de testeo local para tu aplicación.

```
$ sam local <subcommand>
```

Para obtener más información acerca de cada subcomando, consulta lo siguiente:

- [Introducción a sam local generate-event](#)— Genere Servicio de AWS eventos para las pruebas locales.
- [Introducción a sam local invoke](#)— Inicie una única invocación local de una función de AWS Lambda evento.
- [Introducción a sam local start-api](#): ejecuta las funciones de Lambda mediante un servidor HTTP local.
- [Introducción a sam local start-lambda](#)— Ejecute las funciones de Lambda mediante un servidor HTTP local para utilizarlas con los AWS CLI o los SDK.

Introducción a las pruebas con generate-event en local en sam

Utilice el AWS Serverless Application Model `sam local generate-event` subcomando Command Line Interface (AWS SAMCLI) para generar muestras de carga útil de eventos compatibles. Servicios de AWS A continuación, puede modificar estos eventos y transferirlos a los recursos locales para probarlos.

- Para obtener una introducción al, consulte AWS SAMCLI [¿Qué es el AWS SAM CLI?](#)
- Para ver una lista de opciones de comandos `sam local generate-event`, consulta [sam local generate-event](#).

Un evento es un objeto JSON que se genera cuando un Servicio de AWS realiza una acción o tarea. Estos eventos contienen información específica, como los datos que se procesaron o la marca de tiempo del evento. La mayoría de Servicios de AWS generan eventos y los eventos de cada servicio tienen un formato exclusivo para su servicio.

Los eventos generados por un servicio se transfieren a otros servicios como fuente de eventos. Por ejemplo, un artículo colocado en un bucket de Amazon Simple Storage Service (Amazon S3) puede generar un evento. Luego, este evento se puede usar como fuente de eventos para que una función de AWS Lambda evento procese los datos aún más.

Los eventos con `sam local generate-event` los que se generan tienen el mismo formato que los eventos reales creados por el AWS servicio. Puede modificar el contenido de estos eventos y usarlos para probar los recursos de la aplicación.

Requisitos previos

Para usar `sam local generate-event`, completa lo siguiente para instalar la CLI de AWS SAM:

- [AWS SAM requisitos previos](#).
- [Instala la AWS SAM CLI](#).

Antes de usar `sam local generate-event`, se recomienda tener conocimientos básicos de lo siguiente:

- [Configuración de la AWS SAM de CLI](#).
- [Crea tu aplicación en AWS SAM](#).
- [Introducción a la construcción con AWS SAM](#).
- [Introducción a la implementación con AWS SAM](#).

Generar eventos de ejemplo

Utilice el AWS SAMCLI `sam local generate-event` subcomando para generar los eventos compatibles. Servicios de AWS

Para ver una lista de los compatibles Servicios de AWS

1. Ejecuta lo siguiente:

```
$ sam local generate-event
```

2. Aparecerá la lista de compatibles Servicios de AWS . A continuación, se muestra un ejemplo:

```
$ sam local generate-event
...
Commands:
  alb
  alexa-skills-kit
  alexa-smart-home
  apigateway
  appsync
  batch
  cloudformation
  ...
```

Para generar un evento local

1. Ejecute `sam local generate-event` y proporciona el nombre del servicio compatible. Esto mostrará una lista de tipos de eventos que puede generar. A continuación, se muestra un ejemplo:

```
$ sam local generate-event s3

Usage: sam local generate-event s3 [OPTIONS] COMMAND [ARGS]...

Options:
  -h, --help  Show this message and exit.

Commands:
  batch-invocation  Generates an Amazon S3 Batch Operations Invocation Event
  delete            Generates an Amazon S3 Delete Event
  put               Generates an Amazon S3 Put Event
```

2. Para generar el evento de muestra `sam local generate-event`, ejecuta e indique el servicio y el tipo de evento.


```
$ sam local generate-event <service> <event>
```

A continuación, se muestra un ejemplo:

```
$ sam local generate-event s3 put
{
  "Records": [
    {
      "eventVersion": "2.0",
      "eventSource": "aws:s3",
      "awsRegion": "us-east-1",
      "eventTime": "1970-01-01T00:00:00.000Z",
      "eventName": "ObjectCreated:Put",
      "userIdentity": {
        "principalId": "EXAMPLE"
      },
      "requestParameters": {
        "sourceIPAddress": "127.0.0.1"
      },
      "responseElements": {
        "x-amz-request-id": "EXAMPLE123456789",
        "x-amz-id-2": "EXAMPLE123/5678abcdefghijklmbdaisawesome/
mnopqrstuvwxyzABCDEFGH"
      },
      "s3": {
        "s3SchemaVersion": "1.0",
        "configurationId": "testConfigRule",
        "bucket": {
          "name": "sam-s3-demo-bucket",
          "ownerIdentity": {
            "principalId": "EXAMPLE"
          },
          "arn": "arn:aws:s3:::sam-s3-demo-bucket"
        },
        "object": {
          "key": "test/key",
          "size": 1024,
          "eTag": "0123456789abcdef0123456789abcdef",
          "sequencer": "0A1B2C3D4E5F678901"
        }
      }
    }
  ]
}
```

```
]
}
```

Estos eventos de ejemplo contienen valores de marcador de posición. Puede modificar estos valores para que hagan referencia a los recursos reales de la aplicación o a los valores para facilitar las pruebas locales.

Para modificar un evento de muestra

1. Puede modificar eventos de ejemplo en la línea de comandos. Para ver sus opciones, ejecuta lo siguiente:

```
$ sam local generate-event <service> <event> --help
```

A continuación, se muestra un ejemplo:

```
$ sam local generate-event s3 put --help
```

```
Usage: sam local generate-event s3 put [OPTIONS]
```

Options:

<code>--region TEXT</code>	Specify the region name you'd like, otherwise the default = us-east-1
<code>--partition TEXT</code>	Specify the partition name you'd like, otherwise the default = aws
<code>--bucket TEXT</code>	Specify the bucket name you'd like, otherwise the default = example-bucket
<code>--key TEXT</code>	Specify the key name you'd like, otherwise the default = test/key
<code>--debug</code>	Turn on debug logging to print debug message generated by AWS SAM CLI and display timestamps.
<code>--config-file TEXT</code>	Configuration file containing default parameter values. [default: samconfig.toml]
<code>--config-env TEXT</code>	Environment name specifying default parameter values in the configuration file. [default: default]
<code>-h, --help</code>	Show this message and exit.

2. Utiliza cualquiera de estas opciones en la línea de comandos para modificar la carga útil del evento de muestra. A continuación, se muestra un ejemplo:

```
$ sam local generate-event s3 put --bucket sam-s3-demo-bucket
```

```

{
  "Records": [
    {
      "eventVersion": "2.0",
      "eventSource": "aws:s3",
      "awsRegion": "us-east-1",
      "eventTime": "1970-01-01T00:00:00.000Z",
      "eventName": "ObjectCreated:Put",
      "userIdentity": {
        "principalId": "EXAMPLE"
      },
      "requestParameters": {
        "sourceIPAddress": "127.0.0.1"
      },
      "responseElements": {
        "x-amz-request-id": "EXAMPLE123456789",
        "x-amz-id-2": "EXAMPLE123/5678abcdefghijklmbdaisawesome/
mnopqrstuvwxyzABCDEFGH"
      },
      "s3": {
        "s3SchemaVersion": "1.0",
        "configurationId": "testConfigRule",
        "bucket": {
          "name": "sam-s3-demo-bucket",
          "ownerIdentity": {
            "principalId": "EXAMPLE"
          },
          "arn": "arn:aws:s3:::sam-s3-demo-bucket"
        },
        "object": {
          "key": "test/key",
          "size": 1024,
          "eTag": "0123456789abcdef0123456789abcdef",
          "sequencer": "0A1B2C3D4E5F678901"
        }
      }
    }
  ]
}

```

Utiliza los eventos generados para realizar testeo local

Guarde los eventos generados localmente y utiliza otros subcomandos `sam local` para realizar las pruebas.

Para guardar los eventos generados localmente

- Ejecuta lo siguiente:

```
$ sam local generate-event <service> <event> <event-option> > <filename.json>
```

El siguiente es un ejemplo de un evento que se guarda como un archivo `s3.json` en la carpeta `events` de nuestro proyecto.

```
sam-app$ sam local generate-event s3 put --bucket amzn-s3-demo-bucket > events/  
s3.json
```

Para usar un evento generado para realizar testeo local

- Pase el evento con otros subcomandos `sam local` mediante la opción `--event`.

A continuación, se muestra un ejemplo del uso del evento `s3.json` para invocar la función de Lambda de forma local:

```
sam-app$ sam local invoke --event events/s3.json S3JsonLoggerFunction

Invoking src/handlers/s3-json-logger.s3JsonLoggerHandler (nodejs18.x)
Local image is up-to-date
Using local image: public.ecr.aws/lambda/nodejs:18-rapid-x86_64.

Mounting /Users/.../sam-app/.aws-sam/build/S3JsonLoggerFunction as /var/
task:ro,delegated, inside runtime container
START RequestId: f4f45b6d-2ec6-4235-bc7b-495ec2ae0128 Version: $LATEST
END RequestId: f4f45b6d-2ec6-4235-bc7b-495ec2ae0128
REPORT RequestId: f4f45b6d-2ec6-4235-bc7b-495ec2ae0128  Init Duration: 1.23 ms
      Duration: 9371.93 ms      Billed Duration: 9372 ms      Memory Size: 128 MB
      Max Memory Used: 128 MB
```

Más información

Para obtener una lista de todas las opciones `sam local generate-event`, consulta [sam local generate-event](#).

Para ver una demostración del uso de `sam local`, consulta [Desarrollo local con AWS SAM . Probando Nube de AWS recursos de entornos de desarrollo locales](#) en la serie Serverless Land Sessions with SAM en YouTube adelante.

Introducción a las pruebas con invocar `sam local`

Utilice el AWS Serverless Application Model `sam local invoke` subcomando Command Line Interface (AWS SAMCLI) para iniciar una única invocación local de una función de AWS Lambda evento.

- Para obtener una introducción a la, consulte AWS SAMCLI [¿Qué es el AWS SAM CLI?](#)
- Para ver una lista de opciones de comandos `sam local invoke`, consulta [invocar sam local](#).
- Para ver un ejemplo de uso de `sam local invoke` durante un flujo de trabajo de desarrollo típico, consulta [Paso 7: \(opcional\) probar tu aplicación localmente](#).

Note

No se recomienda utilizar las capacidades de invocación local de la CLI de SAM en código que no sea de confianza. Para tener un aislamiento completo del entorno local, ejecute el código directamente en el servicio Lambda.

Note

`sam local invoke` admite funciones duraderas con funciones automáticas de puntos de control y reproducción. Al invocar funciones duraderas de forma local, el estado de ejecución se gestiona automáticamente.

Requisitos previos

Para usar `sam local invoke`, completa lo siguiente para instalar la CLI de AWS SAM:

- [AWS SAM requisitos previos.](#)
- [Instala la AWS SAM CLI.](#)

Antes de usar `sam local invoke`, se recomienda tener conocimientos básicos de lo siguiente:

- [Configuración de la AWS SAM de CLI.](#)
- [Crea tu aplicación en AWS SAM.](#)
- [Introducción a la construcción con AWS SAM.](#)
- [Introducción a la implementación con AWS SAM.](#)

Invocar una función de evento de Lambda localmente

De forma predeterminada, cuando ejecuta `sam local invoke`, la CLI de AWS SAM asume que su directorio de trabajo actual es el directorio raíz del proyecto. Primero la CLI de AWS SAM buscará un archivo `template.[yaml|yml]` dentro de una subcarpeta `.aws-sam`. Si no lo encuentra, la CLI de AWS SAM buscará un archivo `template.[yaml|yml]` en su directorio de trabajo actual.

Para invocar una función de eventos de Lambda localmente

1. Desde el directorio raíz del proyecto, ejecuta lo siguiente:

```
$ sam local invoke <options>
```

2. Si la aplicación contiene más de una función, proporciona el identificador lógico de la función. A continuación, se muestra un ejemplo:

```
$ sam local invoke HelloWorldFunction
```

3. La CLI de AWS SAM compila su función en un contenedor local utilizando Docker. Luego invoca la función y genera la respuesta de su función.

A continuación, se muestra un ejemplo:

```
$ sam local invoke
Invoking app.lambda_handler (python3.9)
Local image is out of date and will be updated to the latest runtime. To skip this,
  pass in the parameter --skip-pull-image
Building
image.....
```

```
Using local image: public.ecr.aws/lambda/python:3.9-rapid-x86_64.

Mounting /Users/.../sam-app/.aws-sam/build/HelloWorldFunction as /var/
task:ro,delegated, inside runtime container
START RequestId: 64bf7e54-5509-4762-a97c-3d740498d3df Version: $LATEST
END RequestId: 64bf7e54-5509-4762-a97c-3d740498d3df
REPORT RequestId: 64bf7e54-5509-4762-a97c-3d740498d3df  Init Duration: 1.09 ms
      Duration: 608.42 ms          Billed Duration: 609 ms Memory Size: 128 MB      Max
      Memory Used: 128 MB
{"statusCode": 200, "body": "{\"message\": \"hello world\"}"}
```

Administración de registros de

Cuando se usa `sam local invoke`, la salida del tiempo de ejecución de la función de Lambda (por ejemplo, los registros) se envía a `stderr` y el resultado de la función de Lambda se envía a `stdout`.

A continuación, se muestra un ejemplo de una función de Lambda básica:

```
def handler(event, context):
    print("some log") # this goes to stderr
    return "hello world" # this goes to stdout
```

Puede guardar estas salidas estándar. A continuación, se muestra un ejemplo:

```
$ sam local invoke 1> stdout.log
...

$ cat stdout.log
"hello world"

$ sam local invoke 2> stderr.log
...

$ cat stderr.log
Invoking app.lambda_handler (python3.9)
Local image is up-to-date
Using local image: public.ecr.aws/lambda/python:3.9-rapid-x86_64.
Mounting /Users/.../sam-app/.aws-sam/build/HelloWorldFunction as /var/
task:ro,delegated, inside runtime container
START RequestId: 0b46e646-3bdf-4b58-8beb-242d00912c46 Version: $LATEST
```

```
some log
END RequestId: 0b46e646-3bdf-4b58-8beb-242d00912c46
REPORT RequestId: 0b46e646-3bdf-4b58-8beb-242d00912c46  Init Duration: 0.91 ms
      Duration: 589.19 ms Billed Duration: 590 ms Memory Size: 128 MB Max Memory Used: 128
      MB
```

Puedes utilizar estos resultados estándar para automatizar aún más sus procesos de desarrollo local.

Opciones

Pase eventos personalizados para invocar la función de eventos de Lambda

Para pasar un evento a la función de Lambda, utiliza la opción `--event`. A continuación, se muestra un ejemplo:

```
$ sam local invoke --event events/s3.json S3JsonLoggerFunction
```

Puede crear eventos con el subcomando `sam local generate-event`. Para obtener más información, consulte [Introducción a las pruebas con generate-event en local en sam](#).

Pase variables de entorno al invocar su función de eventos Lambda

Si la función de Lambda usa variables de entorno, puede pasarlas durante las pruebas locales con la opción `--env-vars`. Esta es una excelente forma de probar una función de Lambda localmente con los servicios de su aplicación que ya están implementados en la nube. A continuación, se muestra un ejemplo:

```
$ sam local invoke --env-vars locals.json
```

Especifica una plantilla o función

Para especificar una plantilla a la que CLI de AWS SAM haga referencia, utiliza la opción `--template`. AWS SAM CLI cargará solo esa AWS SAM plantilla y los recursos a los que apunta.

Para invocar una función de una aplicación o pila anidada, proporciona el identificador lógico de la aplicación o pila junto con el identificador lógico de la función. A continuación, se muestra un ejemplo:

```
$ sam local invoke StackLogicalId/FunctionLogicalId
```


Pruebe una función de evento Lambda desde su Terraform proyecto

Utiliza la opción `--hook-name` para probar localmente las funciones de Lambda de sus proyectos Terraform. Para obtener más información, consulta [Uso de AWS SAM CLI por Terraform para depuración y pruebas locales](#).

A continuación, se muestra un ejemplo:

```
$ sam local invoke --hook-name terraform --beta-features
```

Prácticas recomendadas

Si la aplicación tiene un directorio `.aws-sam` desde el que se ejecuta `sam build`, asegúrese de ejecutar `sam build` cada vez que actualice el código de la función. A continuación, ejecuta `sam local invoke` para probar localmente el código de función actualizado.

El testeo local son una excelente solución para desarrollar y probar rápidamente antes de implementarlas en la nube. Sin embargo, las testeo local no lo validan todo, como los permisos entre los recursos de la nube. En la medida de lo posible, pruebe sus aplicaciones en la nube. Te recomendamos que [utiliza sam sync](#) para acelerar los flujos de trabajo de pruebas en la nube.

Ejemplos

Genere un evento de muestra de Amazon API Gateway y utilícelo para invocar una función de eventos de Lambda de forma local

En primer lugar, generamos una carga útil de eventos API HTTP de API Gateway y la guardamos en la carpeta `events`.

```
$ sam local generate-event apigateway http-api-proxy > events/apigateway_event.json
```

A continuación, modificamos la función de Lambda para que devuelva un valor de parámetro del evento.

```
def lambda_handler(event, context):
    print("HelloWorldFunction invoked")
    return {
        "statusCode": 200,
        "body": json.dumps({
            "message": event['queryStringParameters']['parameter2'],
        })
```

```
}
```

A continuación, invocamos localmente la función de Lambda y proporcionamos un evento personalizado.

```
$ sam local invoke --event events/apigateway_event.json
```

```
Invoking app.lambda_handler (python3.9)
```

```
Local image is up-to-date
```

```
Using local image: public.ecr.aws/lambda/python:3.9-rapid-x86_64.
```

```
Mounting /Users/...sam-app/.aws-sam/build/HelloWorldFunction as /var/task:ro,delegated,
inside runtime container
```

```
START RequestId: 59535d0d-3d9e-493d-8c98-6264e8e961b8 Version: $LATEST
```

```
some log
```

```
END RequestId: 59535d0d-3d9e-493d-8c98-6264e8e961b8
```

```
REPORT RequestId: 59535d0d-3d9e-493d-8c98-6264e8e961b8 Init Duration: 1.63 ms
```

```
Duration: 564.07 ms Billed Duration: 565 ms Memory Size: 128 MB Max Memory
Used: 128 MB
```

```
{"statusCode": 200, "body": "{\"message\": \"value\"}"}
```

Pasar variables de entorno al invocar una función de eventos de Lambda localmente

Esta aplicación tiene una función de Lambda que utiliza una variable de entorno como nombre de tabla de Amazon DynamoDB. El siguiente es un ejemplo de la función definida en la AWS SAM plantilla:

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

```
Transform: AWS::Serverless-2016-10-31
```

```
...
```

```
Resources:
```

```
  getAllItemsFunction:
```

```
    Type: AWS::Serverless::Function
```

```
    Properties:
```

```
      Handler: src/get-all-items.getAllItemsHandler
```

```
      Description: get all items
```

```
      Policies:
```

```
        - DynamoDBReadPolicy:
```

```
          TableName: !Ref SampleTable
```

```
    Environment:
```

```
      Variables:
```

```
        SAMPLE_TABLE: !Ref SampleTable
```

...

Queremos probar localmente la función de Lambda mientras interactúa con la tabla de DynamoDB en la nube. Para ello, creamos un archivo de variables de entorno y lo guardamos en el directorio raíz del proyecto como `locals.json`. El valor se proporciona aquí para las referencias del `SAMPLE_TABLE` de la tabla de DynamoDB en la nube.

```
{
  "getAllItemsFunction": {
    "SAMPLE_TABLE": "dev-demo-SampleTable-1U991234LD5UM98"
  }
}
```

A continuación, ejecutamos `sam local invoke` y transferimos la variables de entorno con la opción `--env-vars`.

```
$ sam local invoke getAllItemsFunction --env-vars locals.json

Mounting /Users/...sam-app/.aws-sam/build/HelloWorldFunction as /var/task:ro,delegated,
inside runtime container
START RequestId: 59535d0d-3d9e-493d-8c98-6264e8e961b8 Version: $LATEST
some log
END RequestId: 59535d0d-3d9e-493d-8c98-6264e8e961b8
REPORT RequestId: 59535d0d-3d9e-493d-8c98-6264e8e961b8  Init Duration: 1.63 ms
  Duration: 564.07 ms      Billed Duration: 565 ms Memory Size: 128 MB      Max Memory
  Used: 128 MB
{"statusCode":200,"body":"{"}]"}}
```

Más información

Para obtener una lista de todas las opciones `sam local invoke`, consulta [invocar sam local](#).

Para ver una demostración del uso de `sam local`, consulta [Desarrollo local con AWS SAM . Probando Nube de AWS recursos de entornos de desarrollo locales](#) en la serie Serverless Land Sessions with SAM en YouTube adelante.

Introducción a las pruebas con `sam local start-api`

Utilice el AWS Serverless Application Model `sam local start-api` subcomando Command Line Interface (AWS SAMCLI) para ejecutar sus funciones de AWS Lambda eventos de forma local y

realizar pruebas en un servidor HTTP local. Este tipo de pruebas resulta útil para las funciones de Lambda que invocan un punto de conexión de Amazon API Gateway.

- Para obtener una introducción al AWS SAMCLI, consulte [¿Qué es el AWS SAM CLI?](#)
- Para ver una lista de opciones de comandos `sam local start-api`, consulta [sam local start-api](#).
- Para ver un ejemplo de uso de `sam local start-api` durante un flujo de trabajo de desarrollo típico, consulta [Paso 7: \(opcional\) probar tu aplicación localmente](#).

Note

`sam local start-api` admite funciones duraderas con funciones automáticas de puntos de control y reproducción. Las funciones duraderas funcionan a la perfección con el punto final API Gateway local.

Requisitos previos

Para usar `sam local start-api`, completa lo siguiente para instalar la CLI de AWS SAM:

- [AWS SAM requisitos previos](#).
- [Instala la AWS SAM CLI](#).

Antes de usar `sam local start-api`, se recomienda tener conocimientos básicos de lo siguiente:

- [Configuración de la AWS SAM de CLI](#).
- [Crea tu aplicación en AWS SAM](#).
- [Introducción a la construcción con AWS SAM](#).
- [Introducción a la implementación con AWS SAM](#).

Uso de una API de inicio local

De forma predeterminada, cuando ejecuta `sam local start-api`, la CLI de AWS SAM asume que su directorio de trabajo actual es el directorio raíz del proyecto. Primero la CLI de AWS SAM buscará un archivo `template.[yaml|yml]` dentro de una subcarpeta `.aws-sam`. Si no lo

encuentra, la CLI de AWS SAM buscará un archivo `template.[yaml|yml]` en su directorio de trabajo actual.

Para iniciar un servidor HTTP local

1. Desde el directorio raíz del proyecto, ejecuta lo siguiente:

```
$ sam local start-api <options>
```

2. La CLI de AWS SAM compila las funciones de Lambda en un contenedor local Docker. A continuación, genera la dirección local del punto de conexión del servidor HTTP. A continuación, se muestra un ejemplo:

```
$ sam local start-api
```

```
Initializing the lambda functions containers.
```

```
Local image is up-to-date
```

```
Using local image: public.ecr.aws/lambda/python:3.9-rapid-x86_64.
```

```
Mounting /Users/.../sam-app/.aws-sam/build/HelloWorldFunction as /var/  
task:ro,delegated, inside runtime container
```

```
Containers Initialization is done.
```

```
Mounting HelloWorldFunction at http://127.0.0.1:3000/hello [GET]
```

```
You can now browse to the above endpoints to invoke your functions. You do not  
need to restart/reload SAM CLI while working on your functions, changes will be  
reflected instantly/automatically. If you used sam build before running local  
commands, you will need to re-run sam build for the changes to be picked up. You  
only need to restart SAM CLI if you update your AWS SAM template
```

```
2023-04-12 14:41:05 WARNING: This is a development server. Do not use it in a  
production deployment. Use a production WSGI server instead.
```

```
* Running on http://127.0.0.1:3000
```

3. Puedes invocar la función de Lambda mediante el navegador o la línea de comandos. A continuación, se muestra un ejemplo:

```
sam-app$ curl http://127.0.0.1:3000/hello  
{ "message": "Hello world!" }%
```

4. Al realizar cambios en el código de la función de Lambda, tenga en cuenta lo siguiente para actualizar el servidor HTTP local:

- Si la aplicación no tiene un directorio `.aws-sam` y la función utiliza un lenguaje interpretado, la CLI de AWS SAM actualizará automáticamente la función creando un nuevo contenedor y alojándolo.
- Si la aplicación tiene un directorio `.aws-sam`, debe ejecutar `sam build` para actualizar la función. A continuación, ejecuta nuevamente `sam local start-api` para alojar la función.
- Si la función utiliza un lenguaje compilado o si el proyecto requiere un soporte de empaquetado complejo, ejecuta su propia solución de compilación para actualizar la función. A continuación, ejecuta nuevamente `sam local start-api` para alojar la función.

Funciones de eventos de Lambda que utilizan autorizadores de Lambda

Note

Esta característica es nueva en la versión 1.80.0 de la CLI de AWS SAM. Para actualizar, consulta [Actualización de AWS SAM CLI](#).

En el caso de las funciones de Lambda configuradas para utilizar autorizadores de Lambda, la CLI de AWS SAM invocará automáticamente la función de Lambda antes de invocar el punto de conexión de la función de Lambda.

A continuación, se muestra un ejemplo de cómo iniciar un servidor HTTP local para una función que utiliza un autorizador Lambda:

```
$ sam local start-api
```

```
2023-04-17 15:02:13 Attaching import module proxy for analyzing dynamic imports
```

```
AWS SAM CLI does not guarantee 100% fidelity between authorizers locally  
and authorizers deployed on AWS. Any application critical behavior should  
be validated thoroughly before deploying to production.
```

```
Testing application behaviour against authorizers deployed on AWS can be done using the  
sam sync command.
```

```
Mounting HelloWorldFunction at http://127.0.0.1:3000/authorized-request [GET]
```

```
You can now browse to the above endpoints to invoke your functions. You do not need  
to restart/reload SAM CLI while working on your functions, changes will be reflected  
instantly/automatically. If you used sam build before running local commands, you will
```

```

need to re-run sam build for the changes to be picked up. You only need to restart SAM
CLI if you update your AWS SAM template
2023-04-17 15:02:13 WARNING: This is a development server. Do not use it in a
production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:3000
2023-04-17 15:02:13 Press CTRL+C to quit

```

Al invocar el punto de conexión de la función de Lambda a través del servidor HTTP local, el primero invoca el autorizador Lambda de la CLI de AWS SAM. Si la autorización se realiza correctamente, la CLI de AWS SAM invocará el punto de conexión de la función de Lambda. A continuación, se muestra un ejemplo:

```

$ curl http://127.0.0.1:3000/authorized-request --header "header:my_token"
{"message": "from authorizer"}%

Invoking app.authorizer_handler (python3.8)
Local image is up-to-date
Using local image: public.ecr.aws/lambda/python:3.8-rapid-x86_64.

Mounting /Users/.../sam-app/... as /var/task:ro,delegated, inside runtime container
START RequestId: 38d3b472-a2c8-4ea6-9a77-9b386989bef0 Version: $LATEST
END RequestId: 38d3b472-a2c8-4ea6-9a77-9b386989bef0
REPORT RequestId: 38d3b472-a2c8-4ea6-9a77-9b386989bef0      Init Duration: 1.08 ms
      Duration: 628.26 msBilled Duration: 629 ms      Memory Size: 128 MB      Max Memory Used:
      128 MB
Invoking app.request_handler (python3.8)
Using local image: public.ecr.aws/lambda/python:3.8-rapid-x86_64.

Mounting /Users/.../sam-app/... as /var/task:ro,delegated, inside runtime container
START RequestId: fdc12255-79a3-4365-97e9-9459d06446ff Version: $LATEST
END RequestId: fdc12255-79a3-4365-97e9-9459d06446ff
REPORT RequestId: fdc12255-79a3-4365-97e9-9459d06446ff      Init Duration: 0.95 ms
      Duration: 659.13 msBilled Duration: 660 ms      Memory Size: 128 MB      Max Memory Used:
      128 MB
No Content-Type given. Defaulting to 'application/json'.
2023-04-17 15:03:03 127.0.0.1 - - [17/Apr/2023 15:03:03] "GET /authorized-request
HTTP/1.1" 200 -

```

Opciones

Reutiliza los contenedores de forma continua para acelerar las invocaciones de funciones locales

De forma predeterminada, la CLI de AWS SAM crea un contenedor nuevo cada vez que se invoca la función a través del servidor HTTP local. Utiliza la opción `--warm-containers` de reutilizar automáticamente el contenedor para las invocaciones de funciones. Esto acelera el tiempo que tarda la CLI de AWS SAM en preparar la función de Lambda para la invocación local. Puede personalizar aún más esta opción proporcionando el argumento `eager` o `lazy`.

- `eager`: los contenedores de todas las funciones se cargan al inicio y se conservan entre las invocaciones.
- `lazy`: los contenedores solo se cargan cuando se invoca cada función por primera vez. A continuación, se conservan para realizar más invocaciones.

A continuación, se muestra un ejemplo:

```
$ sam local start-api --warm-containers eager
```

Al utilizar `--warm-containers` y modificar el código de la función de Lambda:

- Si la aplicación tiene un directorio `.aws-sam`, ejecuta `sam build` para actualizar el código de función en los artefactos de compilación de la aplicación.
- Cuando se detecta un cambio en el código, la CLI de AWS SAM cierra automáticamente el contenedor de funciones de Lambda.
- Cuando vuelva a invocar la función, la CLI de AWS SAM crea automáticamente un nuevo contenedor.

Si la supervisión de archivos provoca una I/O sobrecarga o sobrecarga de la CPU (por ejemplo, en monorepos grandes, entornos con software antivirus o proyectos en los que varias funciones comparten una `CodeUri` función), puede desactivarla con la siguiente opción: `--no-watch`

```
$ sam local start-api --warm-containers eager --no-watch
```

Con `--no-watch`, los cambios en el código y la plantilla no se detectan automáticamente. Detenga y vuelva a ejecutar el comando para recoger los cambios.

Especifique una imagen de contenedor para utilizarla en las funciones de eventos de Lambda

De forma predeterminada, la CLI de AWS SAM utiliza imágenes base de Lambda de Amazon Elastic Container Registry (Amazon ECR) para invocar sus funciones de forma local. Utiliza la opción `--invoke-image` para hacer referencia a una imagen de contenedor personalizada. A continuación, se muestra un ejemplo:

```
$ sam local start-api --invoke-image public.ecr.aws/sam/emu-python3.8
```

Puedes especificar la función que se va a utilizar con la imagen de contenedor personalizada. A continuación, se muestra un ejemplo:

```
$ sam local start-api --invoke-image Function1=amazon/aws/sam-cli-emulation-image-python3.8
```

Especifica una plantilla para probarla localmente

Para especificar una plantilla a la que CLI de AWS SAM haga referencia, utiliza la opción `--template`. AWS SAMCLICargará solo esa AWS SAM plantilla y los recursos a los que apunta. A continuación, se muestra un ejemplo:

```
$ sam local start-api --template myTemplate.yaml
```

Especifique el entorno de desarrollo anfitrión de su función de eventos Lambda

De forma predeterminada, el subcomando `sam local start-api` crea un servidor HTTP localhost con una dirección IP 127.0.0.1. Puede personalizar estos valores si su entorno de desarrollo local está aislado de su máquina local.

Utiliza la opción `--container-host` para especificar un host. A continuación, se muestra un ejemplo:

```
$ sam local start-api --container-host host.docker.internal
```

Utiliza la opción `--container-host-interface` para especificar un la dirección IP del host de la red a la que deben enlazarse los puertos del contenedor. A continuación, se muestra un ejemplo:

```
$ sam local start-api --container-host-interface 0.0.0.0
```

Prácticas recomendadas

Si la aplicación tiene un directorio `.aws-sam` desde el que se ejecuta `sam build`, asegúrese de ejecutar `sam build` cada vez que actualice el código de la función. A continuación, ejecuta `sam local start-api` para probar localmente el código de función actualizado.

El testeo local son una excelente solución para desarrollar y probar rápidamente antes de implementarlas en la nube. Sin embargo, las testeo local no lo validan todo, como los permisos entre los recursos de la nube. En la medida de lo posible, pruebe sus aplicaciones en la nube. Te recomendamos que [utiliza sam sync](#) para acelerar los flujos de trabajo de pruebas en la nube.

Más información

Para obtener una lista de todas las opciones `sam local start-api`, consulta [sam local start-api](#).

Introducción a las pruebas con sam local start-lambda

Utilice el AWS SAMCLI subcomando `sam local start-lambda` para invocar la función de eventos de Lambda a través de los SDK y. AWS CLI Este comando inicia un punto de conexión local que emula Lambda.

- Para obtener una introducción a la, consulte AWS SAMCLI [¿Qué es el AWS SAM CLI?](#)
- Para ver una lista de opciones de comandos `sam local start-lambda`, consulta [sam local start-lambda](#).

Note

`sam local start-lambda` admite funciones duraderas con funciones automáticas de puntos de control y reproducción. Las funciones duraderas funcionan a la perfección con el punto final Lambda local.

Requisitos previos

Para usar `sam local start-lambda`, completa lo siguiente para instalar la CLI de AWS SAM:

- [AWS SAM requisitos previos](#).
- [Instala la AWS SAM CLI](#).

Antes de usar `sam local start-lambda`, se recomienda tener conocimientos básicos de lo siguiente:

- [Configuración de la AWS SAM de CLI.](#)
- [Crea tu aplicación en AWS SAM.](#)
- [Introducción a la construcción con AWS SAM.](#)
- [Introducción a la implementación con AWS SAM.](#)

Utilizando `sam local start-lambda`

De forma predeterminada, cuando ejecuta `sam local start-lambda`, la CLI de AWS SAM asume que su directorio de trabajo actual es el directorio raíz del proyecto. Primero la CLI de AWS SAM buscará un archivo `template.[yaml|yml]` dentro de una subcarpeta `.aws-sam`. Si no lo encuentra, la CLI de AWS SAM buscará un archivo `template.[yaml|yml]` en su directorio de trabajo actual.

Para usar `sam local start-lambda`

1. Desde el directorio raíz del proyecto, ejecuta lo siguiente:

```
$ sam local start-lambda <options>
```

2. La CLI de AWS SAM compila las funciones de Lambda en un contenedor local Docker. A continuación, envía la dirección local al punto de conexión de su servidor HTTP. A continuación, se muestra un ejemplo:

```
$ sam local start-lambda
Initializing the lambda functions containers.
Local image is up-to-date
Using local image: public.ecr.aws/lambda/python:3.9-rapid-x86_64.

Mounting /Users/.../sam-app/hello_world as /var/task:ro,delegated, inside runtime
container
Containers Initialization is done.
Starting the Local Lambda Service. You can now invoke your Lambda Functions defined
in your template through the endpoint.
2023-04-13 07:25:43 WARNING: This is a development server. Do not use it in a
production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:3001
```

```
2023-04-13 07:25:43 Press CTRL+C to quit
```

3. Utilice los AWS CLI o los SDK para invocar la función Lambda de forma local.

A continuación, se muestra un ejemplo mediante la AWS CLI:

```
$ aws lambda invoke --function-name "HelloWorldFunction" --endpoint-url "http://127.0.0.1:3001" --no-verify-ssl out.txt
```

```
StatusCode: 200
```

```
(END)
```

A continuación, se muestra un ejemplo en el que se utiliza el for: AWS SDK Python

```
import boto3
from botocore.config import Config
from botocore import UNSIGNED

lambda_client = boto3.client('lambda',
                              endpoint_url="http://127.0.0.1:3001",
                              use_ssl=False,
                              verify=False,
                              config=Config(signature_version=UNSIGNED,
                                              read_timeout=1,
                                              retries={'max_attempts': 0})
                              )

lambda_client.invoke(FunctionName="HelloWorldFunction")
```

Opciones

Especificar plantilla

Para especificar una plantilla a la que CLI de AWS SAM haga referencia, utiliza la opción `--template`. AWS SAM CLI cargará solo esa AWS SAM plantilla y los recursos a los que apunta. A continuación, se muestra un ejemplo:

```
$ sam local start-lambda --template myTemplate.yaml
```

Para obtener más información sobre AWS SAM las plantillas, consulte [AWS SAM anatomía de la plantilla](#).

Prácticas recomendadas

Si la aplicación tiene un directorio `.aws-sam` desde el que se ejecuta `sam build`, asegúrese de ejecutar `sam build` cada vez que actualice el código de la función. A continuación, ejecuta `sam local start-lambda` para probar localmente el código de función actualizado.

El testeo local son una excelente solución para desarrollar y probar rápidamente antes de implementarlas en la nube. Sin embargo, las testeo local no lo validan todo, como los permisos entre los recursos de la nube. En la medida de lo posible, pruebe sus aplicaciones en la nube. Te recomendamos que [utiliza sam sync](#) para acelerar los flujos de trabajo de pruebas en la nube.

Más información

Para obtener una lista de todas las opciones `sam local start-lambda`, consulta [sam local start-lambda](#).

Invoque localmente funciones de eventos de Lambda con AWS SAM

La invocación local de una función de eventos de Lambda antes de probarla o implementarla en la nube puede tener diversas ventajas. Le permite probar la lógica de la función con mayor rapidez. Probar primero a nivel local reduce la probabilidad de identificar problemas al realizar las pruebas en la nube o durante la implementación, lo que puede ayudarle a evitar costos innecesarios. Además, las pruebas locales facilitan la depuración.

Puede invocar la función de Lambda de forma local mediante el comando [invocar sam local](#) y proporcionando el ID lógico de la función y un archivo de eventos. `sam local invoke` también acepta `stdin` como evento. Para más información, consulta [Evento](#) en la Guía para desarrolladores de AWS Lambda . Para obtener información sobre los formatos de mensajes de eventos de diferentes AWS servicios, consulte [Utilización AWS Lambda con otros servicios](#) en la Guía para AWS Lambda desarrolladores.

Note

No se recomienda utilizar las capacidades de invocación local de la CLI de SAM en código que no sea de confianza. Para tener un aislamiento total del entorno local, ejecute el código directamente en el servicio Lambda.

Note

El `sam local invoke` comando corresponde al comando AWS Command Line Interface [aws lambda invoke](#) (AWS CLI). Puedes utilizar cualquiera de los comandos para invocar una función de Lambda.

Debes ejecutar el comando `sam local invoke` en el directorio del proyecto que contiene la función que desea invocar.

Ejemplos:

```
# Invoking function with event file
$ sam local invoke "Ratings" -e event.json

# Invoking function with event via stdin
$ echo '{"message": "Hey, are you there?" }' | sam local invoke --event - "Ratings"

# For more options
$ sam local invoke --help
```

Archivo de variables de entorno

Para declarar localmente las variables de entorno que anulan los valores definidos en las plantillas, haga lo siguiente:

1. Cree un JSON o un `.env` archivo que contenga las variables de entorno que desee anular.
2. Usa el argumento `--env-vars` para anular los valores definidos en sus plantillas.

La `--env-vars` opción admite dos formatos de archivo. El formato del archivo se detecta automáticamente en función del contenido del archivo.

Declarar variables de entorno con JSON

Para declarar variables de entorno que se apliquen globalmente a todos los recursos, especifica un objeto `Parameters` como el siguiente:

```
{
  "Parameters": {
```

```
    "TABLE_NAME": "localtable",
    "BUCKET_NAME": "amzn-s3-demo-bucket",
    "STAGE": "dev"
  }
}
```

Para declarar variables de entorno diferentes para cada recurso, especifica objetos para cada recurso de la siguiente manera:

```
{
  "MyFunction1": {
    "TABLE_NAME": "localtable",
    "BUCKET_NAME": "amzn-s3-demo-bucket",
  },
  "MyFunction2": {
    "TABLE_NAME": "localtable",
    "STAGE": "dev"
  }
}
```

Al especificar objetos para cada recurso, puede utilizar los siguientes identificadores, enumerados en orden de mayor a menor prioridad:

1. `logical_id`
2. `function_id`
3. `function_name`
4. Identificador de ruta completo

Puedes utilizar los dos métodos anteriores para declarar variables de entorno juntos en un único archivo. Al hacerlo, las variables de entorno que haya proporcionado para recursos específicos tienen prioridad sobre las variables de entorno globales.

Guarde las variables de entorno en un archivo JSON, como `env.json`.

Declarar variables de entorno con `archivos.env`

También puede usar un `.env` archivo para declarar variables de entorno. Las variables declaradas en un `.env` archivo se aplican globalmente a todas las funciones, lo que equivale al `Parameters` objeto en formato JSON.

```
TABLE_NAME=localtable
BUCKET_NAME=amzn-s3-demo-bucket
STAGE=dev
```

El `.env` formato admite comentarios (líneas que comienzan por `#`) y valores entre comillas.

Note

El `.env` formato solo admite variables de entorno globales. Para declarar variables de entorno específicas de una función, utilice el formato JSON.

Anulación de valores de variables de entorno

Para sustituir las variables de entorno por las definidas en el archivo de variables de entorno, utilice el `--env-vars` argumento con los `invoke` comandos o `start-api`. Por ejemplo:

```
# Using a JSON file
sam local invoke --env-vars env.json

# Using a .env file
sam local invoke --env-vars .env
```

Capas

Si tu aplicación incluye capas, para obtener información sobre cómo depurar los problemas con las capas de su host local, consulta [Aumente la eficiencia con las capas Lambda con AWS SAM](#).

Más información

Para ver un ejemplo práctico de cómo invocar funciones localmente, consulte el [Módulo 2: Ejecutar localmente](#) en The Complete Workshop. AWS SAM

Ejecute API Gateway de forma local con AWS SAM

La ejecución local de Amazon API Gateway puede tener diversas ventajas. Por ejemplo, ejecutar API Gateway de forma local te permite probar los puntos finales de la API de forma local antes de implementarlos AWS en la nube. Si primero realiza las pruebas a nivel local, a menudo puede

reducir las pruebas y el desarrollo en la nube, lo que puede ayudar a reducir los costos. Además, las pruebas locales facilitan la depuración.

Para iniciar una instancia local de API Gateway que puedas usar para probar la funcionalidad de solicitud/respuesta HTTP, usa el comando `aws-sam local start-api`. Esta funcionalidad incluye la recarga en caliente para que pueda desarrollar e iterar rápidamente sus características.

Note

La recarga en caliente se produce cuando solo se actualizan los archivos que se han modificado y el estado de la aplicación sigue siendo el mismo. Por el contrario, la recarga en tiempo real se produce cuando se actualiza toda la aplicación y se pierde su estado.

Para obtener instrucciones sobre cómo utilizar el comando `aws-sam local start-api`, consulta [Introducción a las pruebas con aws-sam local start-api](#).

De forma predeterminada, AWS SAM usa integraciones de AWS Lambda proxy y es compatible con ambos tipos de recursos. Para obtener más información sobre las integraciones de proxy para los tipos de recursos, consulte [Cómo trabajar con integraciones de AWS Lambda proxy para HTTP APIs](#) en la Guía para desarrolladores de API Gateway. Para obtener más información sobre las integraciones de proxy para los tipos de recursos, consulte [Cómo trabajar con integraciones de proxy Lambda de API Gateway](#) en la Guía para desarrolladores de API Gateway.

Ejemplo:

```
$ aws-sam local start-api
```

AWS SAM busca automáticamente todas las funciones de la AWS SAM plantilla que tengan definidas las fuentes de eventos HTTP o las fuentes de eventos. A continuación, monta la función en las rutas HTTP definidas.

En el ejemplo de API siguiente, la función `ratings.py:handler()` en `/ratings` para las solicitudes GET.

```
Ratings:
  Type: AWS::Serverless::Function
```

```
Properties:
  Handler: ratings.handler
  Runtime: python3.9
  Events:
    Api:
      Type: Api
      Properties:
        Path: /ratings
        Method: get
```

Aquí tienes un ejemplo de Api de respuesta:

```
// Example of a Proxy Integration response
exports.handler = (event, context, callback) => {
  callback(null, {
    statusCode: 200,
    headers: { "x-custom-header" : "my custom header value" },
    body: "hello world"
  });
}
```

Si modifica el código de la función, ejecuta el comando `sam build` para `sam local start-api` para detectar los cambios.

Archivo de variables de entorno

Para declarar localmente las variables de entorno que anulan los valores definidos en las plantillas, haga lo siguiente:

1. Cree un archivo JSON que contenga las variables de entorno que desee anular.
2. Usa el argumento `--env-vars` para anular los valores definidos en sus plantillas.

Declaración de variables de entorno

Para declarar variables de entorno que se apliquen globalmente a todos los recursos, especifica un objeto `Parameters` como el siguiente:

```
{
  "Parameters": {
    "TABLE_NAME": "localtable",
```

```
    "BUCKET_NAME": "amzn-s3-demo-bucket",  
    "STAGE": "dev"  
  }  
}
```

Para declarar variables de entorno diferentes para cada recurso, especifica objetos para cada recurso de la siguiente manera:

```
{  
  "MyFunction1": {  
    "TABLE_NAME": "localtable",  
    "BUCKET_NAME": "amzn-s3-demo-bucket",  
  },  
  "MyFunction2": {  
    "TABLE_NAME": "localtable",  
    "STAGE": "dev"  
  }  
}
```

Al especificar objetos para cada recurso, puede utilizar los siguientes identificadores, enumerados en orden de mayor a menor prioridad:

1. `logical_id`
2. `function_id`
3. `function_name`
4. Identificador de ruta completo

Puedes utilizar los dos métodos anteriores para declarar variables de entorno juntos en un único archivo. Al hacerlo, las variables de entorno que haya proporcionado para recursos específicos tienen prioridad sobre las variables de entorno globales.

Guarde las variables de entorno en un archivo JSON, como `env.json`.

Anulación de valores de variables de entorno

Para anular las variables de entorno por las definidas en el archivo JSON, utiliza el argumento `--env-vars` con los comandos `invoke` o `start-api`. Por ejemplo:

```
$ sam local start-api --env-vars env.json
```

Capas

Si tu aplicación incluye capas, para obtener información sobre cómo depurar los problemas con las capas de su host local, consulta [Aumente la eficiencia con las capas Lambda con AWS SAM](#).

Introducción a las pruebas en la nube con evento de prueba remoto de sam

Utilice el AWS Serverless Application Model comando Command Line Interface (AWS SAM CLI) `sam remote test-event` para acceder y gestionar los eventos de prueba que se pueden compartir para sus AWS Lambda funciones de eventos.

Para obtener más información sobre los eventos de prueba compartibles, consulta [Eventos de prueba compartibles](#) en la Guía para desarrolladores de AWS Lambda .

Temas

- [Configure el AWS SAM CLI para usar evento de prueba remoto de sam](#)
- [Uso de evento de prueba remoto de sam comando](#)
- [Uso de eventos de prueba compartibles](#)
- [Gestion de eventos de prueba compartibles](#)

Requisitos previos

Para usar `sam remote test-event`, completa lo siguiente para instalar la CLI de AWS SAM:

- [AWS SAM requisitos previos](#).
- [Instala la AWS SAM CLI](#).

Si ya la tiene AWS SAM CLI instalada, le recomendamos que la actualice a la última AWS SAMCLI versión. Para obtener más información, consulta [Actualización de AWS SAM CLI](#).

Antes de usar `sam remote test-event`, se recomienda tener conocimientos básicos de lo siguiente:

- [Configuración de la AWS SAM de CLI](#).
- [Crea tu aplicación en AWS SAM](#).

- [Introducción a la construcción con AWS SAM.](#)
- [Introducción a la implementación con AWS SAM.](#)
- [Introducción al uso sam sync sincronizar con Nube de AWS.](#)

Configure el AWS SAM CLI para usar evento de prueba remoto de sam

Complete los siguientes pasos de configuración para usar el AWS SAM CLI `sam remote test-event` comando:

1. Configure el AWS SAM CLI para usar su Cuenta de AWS: los usuarios pueden acceder a los eventos de prueba de Lambda que se pueden compartir y administrarlos dentro del mismo. Cuenta de AWS Para configurar los AWS SAM CLI para que usen los suyos Cuenta de AWS, consulte. [Configuración de la AWS SAM de CLI](#)
2. Configurar los permisos para los eventos de prueba compartibles: para acceder a los eventos de prueba compartibles y gestionarlos, debe tener los permisos adecuados. Para obtener más información, consulta los [Eventos de prueba que se puedan compartir](#) en la Guía para desarrolladores de AWS Lambda .

Uso de evento de prueba remoto de sam comando

El AWS SAM CLI `sam remote test-event` comando proporciona los siguientes subcomandos que puede usar para acceder a los eventos de prueba que se pueden compartir y administrarlos:

- `delete`— Eliminar un evento de prueba que se pueda compartir del registro de EventBridge esquemas de Amazon.
- `get`— Obtenga un evento de prueba que se pueda compartir del registro de EventBridge esquemas.
- `list`— Listar los eventos de prueba que se pueden compartir existentes para una función del registro del EventBridge esquema.
- `put`— Guarda un evento de un archivo local en el registro del EventBridge esquema.

Para enumerar estos subcomandos mediante el AWS SAM CLI, ejecute lo siguiente:

```
$ sam remote test-event --help
```

Eliminación de eventos de prueba compartibles

Puede eliminar un evento de prueba compatible usando el subcomando `delete` junto con lo siguiente:

- Proporciona el nombre del evento de prueba compatible que se va a eliminar.
- Proporciona un ID aceptable de la función de Lambda asociada al evento.
- Si proporciona el identificador lógico de la función Lambda, también debe proporcionar el nombre de la AWS CloudFormation pila asociada a la función Lambda.

A continuación, se muestra un ejemplo:

```
$ sam remote test-event delete HelloWorldFunction --stack-name sam-app --name demo-event
```

Para obtener una lista de las opciones que se pueden usar con el subcomando `delete`, consulta [eliminación remota del evento de prueba sam](#). También puede ejecutar lo siguiente desde: AWS SAM CLI

```
$ sam remote test-event delete --help
```

Obtención de eventos de prueba compartibles

Puede obtener un evento de prueba que se pueda compartir desde el registro del EventBridge esquema mediante el `get` subcomando junto con lo siguiente:

- Proporciona el nombre del evento de prueba compatible que se desea obtener.
- Proporciona un ID aceptable de la función de Lambda asociada al evento.
- Si proporciona el identificador lógico de la función Lambda, también debe proporcionar el nombre de la AWS CloudFormation pila asociada a la función Lambda.

A continuación se muestra un ejemplo en el que se obtiene un evento de prueba compatible llamado `demo-event` que está asociado a la función de Lambda `HelloWorldFunction` de la pila de `sam-app`. Este comando imprimirá el evento en la consola.

```
$ sam remote test-event get HelloWorldFunction --stack-name sam-app --name demo-event
```

Para obtener un evento de prueba compatible y guardarlo en su equipo local, utiliza la opción `--output-file` y proporciona una ruta de archivo y un nombre. A continuación se muestra un ejemplo que guarda `demo-event` como `demo-event.json` en el directorio de trabajo actual:

```
$ sam remote test-event get HelloWorldFunction --stack-name sam-app --name demo-event --output-file demo-event.json
```

Para obtener una lista de las opciones que se pueden usar con el subcomando `get`, consulta [sam remote test-event get](#). También puede ejecutar lo siguiente desde: AWS SAM CLI

```
$ sam remote test-event get --help
```

Enumeración de eventos de prueba compatibles

Puede enumerar todos los eventos de prueba compatibles para una función de Lambda concreta desde el registro de esquemas. Utiliza el subcomando `list` junto con lo siguiente:

- Proporciona un ID aceptable de la función de Lambda asociada a los eventos.
- Si proporciona el identificador lógico de la función Lambda, también debe proporcionar el nombre de la CloudFormation pila asociada a la función Lambda.

A continuación se muestra un ejemplo en el que se obtiene una lista de todos los eventos de prueba compatibles asociados con la función de Lambda `HelloWorldFunction` de la pila de `sam-app`.

```
$ sam remote test-event list HelloWorldFunction --stack-name sam-app
```

Para obtener una lista de las opciones que se pueden usar con el subcomando `list`, consulta [lista de eventos de prueba remotos sam](#). También puede ejecutar lo siguiente desde: AWS SAM CLI

```
$ sam remote test-event list --help
```

Almacenamiento de eventos de prueba compatibles

Puede guardar los eventos de prueba que se pueden compartir en el registro de EventBridge esquemas. Utiliza el subcomando `put` junto con lo siguiente:

- Proporciona un identificador aceptable de la función de Lambda asociada al evento de prueba compatible.

- Proporciona un nombre para el evento de prueba compatible.
- Proporciona la ruta del archivo y el nombre del evento local que desee cargar.

El siguiente es un ejemplo que guarda el evento `demo-event.json` local como `demo-event` y lo asocia a la función de Lambda `HelloWorldFunction` de la pila de `sam-app`:

```
$ sam remote test-event put HelloWorldFunction --stack-name sam-app --name demo-event --file demo-event.json
```

Si existe un evento de prueba que se pueda compartir con el mismo nombre en el registro del EventBridge esquema, no AWS SAM CLI lo sobrescribirá. Para sobrescribir, añada la opción `--force` a su comando.

Para obtener una lista de las opciones que se pueden usar con el subcomando `put`, consulta [salida de evento de prueba remota sam](#). También puede ejecutar lo siguiente desde: AWS SAM CLI

```
$ sam remote test-event put --help
```

Uso de eventos de prueba compatibles

Utilice eventos de prueba que se puedan compartir para probar las funciones de Lambda en Nube de AWS el comando `with sam remote invoke the`. Para obtener más información, consulte [Transfiera eventos de prueba compatibles a una función de eventos de Lambda en la nube](#).

Gestion de eventos de prueba compatibles

Este tema contiene ejemplos sobre cómo administrar y usar los eventos de prueba compatibles.

Obtén un evento de prueba compatible, modifíquelo y úselo

Puede obtener un evento de prueba que se pueda compartir del registro del EventBridge esquema, modificarlo localmente y usar el evento de prueba local con la función Lambda en. Nube de AWS A continuación, se muestra un ejemplo:

1. Recuperar el evento de prueba compatible: utiliza el subcomando `sam remote test-event get` para recuperar un evento de prueba compatible para una función de Lambda específica y guárdalo localmente:


```
$ sam remote test-event get HelloWorldFunction --stack-name sam-app --name demo-event --output-file demo-event.json
```

2. Modifica el evento de prueba compatible: utiliza un editor de texto de tu elección para modificar el evento de prueba compatible.
3. Usar el evento de prueba compatible: usa el `sam remote invoke` comando y proporciona la ruta del archivo y el nombre del evento con: `--event-file`

```
$ sam remote invoke HelloWorldFunction --stack-name sam-app --event-file demo-event.json
```

Obtén un evento de prueba compatible, modifíquelo, cárguelo y úselo

Puede obtener un evento de prueba que se pueda compartir del registro del EventBridge esquema, modificarlo localmente y cargarlo. A continuación, puede pasar el evento de prueba compatible directamente a la función de Lambda en la Nube de AWS. A continuación, se muestra un ejemplo:

1. Recuperar el evento de prueba compatible: utiliza el subcomando `sam remote test-event get` para recuperar un evento de prueba compatible para una función de Lambda específica y guárdalo localmente:

```
$ sam remote test-event get HelloWorldFunction --stack-name sam-app --name demo-event --output-file demo-event.json
```

2. Modifica el evento de prueba compatible: utiliza un editor de texto de tu elección para modificar el evento de prueba compatible.
3. Cargue el evento de prueba que se puede compartir: utilice el `sam remote test-event put` subcomando para cargar y guardar el evento de prueba que se puede compartir en el registro del esquema. EventBridge En este ejemplo, utilizamos la opción `--force` para sobrescribir una versión anterior de nuestra prueba compatible.

```
$ sam remote test-event put HelloWorldFunction --stack-name sam-app --name demo-event --file demo-event.json --force
```

4. Pase el evento de prueba compatible a la función de Lambda: utiliza el comando `sam remote invoke` para pasar el evento de prueba compatible directamente a la función de Lambda en la Nube de AWS:

```
$ sam remote invoke HelloWorldFunction --stack-name sam-app --test-event-name demo-event
```

Introducción a las pruebas en la nube con invocación remota de sam

Utilice el AWS Serverless Application Model comando Command Line Interface (AWS SAM CLI) `sam remote invoke` para interactuar con AWS los recursos compatibles en Nube de AWS. Puedes utilizar `sam remote invoke` para invocar los recursos siguientes:

- Amazon Kinesis Data Streams: envía registros de datos a las aplicaciones de Kinesis Data Streams.
- AWS Lambda— Invoque y pase eventos a sus funciones de eventos de Lambda.
- Amazon Simple Queue Service (Amazon SQS): envía mensajes a colas de Amazon SQS.
- AWS Step Functions: invoca máquinas de estado de Step Functions para iniciar la ejecución.

Para obtener una introducción a las, consulte AWS SAMCLI [¿Qué es el AWS SAM CLI?](#)

Para ver un ejemplo de uso de `sam remote invoke` durante un flujo de trabajo de desarrollo típico, consulta [Paso 5: Interactúa con tu función en el Nube de AWS](#).

Temas

- [Uso del comando `sam remote invoke`](#)
- [Uso de las opciones del comando `sam remote invoke`](#)
- [Configura el archivo de configuración de su proyecto](#)
- [Ejemplos](#)
- [Vínculos relacionados](#)

Requisitos previos

Para usar `sam remote invoke`, completa lo siguiente para instalar la CLI de AWS SAM:

- [AWS SAM requisitos previos](#).
- [Instala la AWS SAM CLI](#).

También recomendamos actualizar a la última versión del AWS SAMCLI. Para obtener más información, consulta [Actualización de AWS SAM CLI](#).

Antes de usar `sam remote invoke`, se recomienda tener conocimientos básicos de lo siguiente:

- [Configuración de la AWS SAM de CLI](#).
- [Crea tu aplicación en AWS SAM](#).
- [Introducción a la construcción con AWS SAM](#).
- [Introducción a la implementación con AWS SAM](#).
- [Introducción al uso `sam sync` sincronizar con Nube de AWS](#).

Uso del comando `sam remote invoke`

Antes de usar este comando, el recurso debe implementarse en la Nube de AWS.

Usa la siguiente estructura de comandos y ejecútela desde el directorio raíz del proyecto:

```
$ sam remote invoke <arguments> <options>
```

Note

Esta página mostrará las opciones que se proporcionan en el símbolo del sistema. También puede configurar las opciones en el archivo de configuración del proyecto en lugar de pasarlas por la línea de comandos. Para obtener más información, [Configurar los ajustes del proyecto](#).

Para obtener una descripción de los argumentos `sam remote invoke` y las opciones, consulta [invocación remota de sam](#).

Uso con Kinesis Data Streams

Puede enviar registros de datos a una aplicación de Kinesis Data Streams. Le AWS SAM CLI enviará su registro de datos y le devolverá un identificador de fragmento y un número de secuencia. A continuación, se muestra un ejemplo:

```
$ sam remote invoke KinesisStream --stack-name kinesis-example --event hello-world
```

```
Putting record to Kinesis data stream KinesisStream
```

```
Auto converting value 'hello-world' into JSON '"hello-world"'. If you don't want auto-
conversion, please provide
a JSON string as event
```

```
{
  "ShardId": "shardId-000000000000",
  "SequenceNumber": "49646251411914806775980850790050483811301135051202232322"
}%
```

Para enviar un registro de datos

1. Proporciona un valor de ID de recurso como argumento para tu aplicación Kinesis Data Streams. Para obtener información sobre los ID de recursos válidos, consulta el [ID de recurso](#).
2. Proporciona el registro de datos como un evento para enviarlo a la aplicación Kinesis Data Streams. Puedes proporcionar el evento en la línea de comandos mediante la opción `--event` o desde un archivo mediante `--event-file`. Si no proporciona un evento, AWS SAM CLI envía un evento vacío.

Uso con funciones de eventos de Lambda

Puedes invocar una función de Lambda en la nube y pasar un evento vacío o proporcionar un evento en la línea de comandos o desde un archivo. AWS SAM CLI invoca la función Lambda y devolverá su respuesta. A continuación, se muestra un ejemplo:

```
$ sam remote invoke HelloWorldFunction --stack-name sam-app
```

```
Invoking Lambda Function HelloWorldFunction
```

```
START RequestId: d5ef494b-5f45-4086-86fd-d7322fa1a1f9 Version: $LATEST
```

```
END RequestId: d5ef494b-5f45-4086-86fd-d7322fa1a1f9
```

```
REPORT RequestId: d5ef494b-5f45-4086-86fd-d7322fa1a1f9 Duration: 6.62 ms      Billed
Duration: 7 ms      Memory Size: 128 MB      Max Memory Used: 67 MB      Init Duration:
164.06 ms
```

```
{"statusCode":200,"body":{"\"message\":\"hello world\""}}%
```

Para invocar una función de eventos de Lambda

1. Proporciona un valor de ID de recurso como argumento para su función de Lambda. Para obtener información sobre los ID de recursos válidos, consulta el [ID de recurso](#).

2. Proporciona un evento para que lo envíe a su función de Lambda. Puedes proporcionar el evento en la línea de comandos mediante la opción `--event` o desde un archivo mediante `--event-file`. Si no proporciona un evento, AWS SAM CLI envía un evento vacío.

Funciones de eventos Lambda configuradas con transmisión de respuestas

El comando `sam remote invoke` admite funciones de Lambda que están configuradas para transmitir respuestas. Puede configurar una función Lambda para transmitir las respuestas mediante la [FunctionUrlConfig](#) propiedad de sus AWS SAM plantillas. Cuando utiliza `sam remote invoke`, la CLI de AWS SAM detectará automáticamente su configuración de Lambda y la invocará con una transmisión de respuesta.

Para ver un ejemplo, consulta [Invoca una función de eventos de Lambda configurada para transmitir respuestas](#).

Transfiera eventos de prueba compatibles a una función de eventos de Lambda en la nube

Los eventos de prueba que se pueden compartir son aquellos que puede compartir con otros en la misma Cuenta de AWS. Para obtener más información, consulta los [Eventos de prueba que se puedan compartir](#) en la Guía para desarrolladores de AWS Lambda .

Acceso y administración de eventos de prueba que se puedan compartir

Puede utilizar el AWS SAM CLI `sam remote test-event` comando para acceder a los eventos de prueba que se pueden compartir y gestionarlos. Por ejemplo, puede utilizar `sam remote test-event` para hacer lo siguiente:

- Recupera eventos de prueba que se pueden compartir del registro de EventBridge esquemas de Amazon.
- Modifique los eventos de prueba que se pueden compartir de forma local y cárguelos en el registro de EventBridge esquemas.
- Elimine los eventos de prueba que se puedan compartir del registro de EventBridge esquemas.

Para obtener más información, consulte [Introducción a las pruebas en la nube con evento de prueba remoto de sam](#).

Transfiera un evento de prueba que se pueda compartir a una función de eventos de Lambda en la nube

Para pasar un evento de prueba que se pueda compartir del registro de EventBridge esquemas a la función Lambda en la nube, utilice la opción y proporcione `--test-event-name` el nombre del evento de prueba que se puede compartir. A continuación, se muestra un ejemplo:

```
$ sam remote invoke HelloWorldFunction --stack-name sam-app --test-event-name demo-event
```

Si guarda el evento de prueba que se puede compartir de forma local, puede usar la opción `--event-file` y proporcionar la ruta del archivo y el nombre del evento de prueba local. A continuación, se muestra un ejemplo:

```
$ sam remote invoke HelloWorldFunction --stack-name sam-app --event-file demo-event.json
```

Uso de con Amazon SQS

Puede enviar mensajes a las colas de Amazon SQS. AWS SAM CLI devuelve lo siguiente:

- Message ID
- MD5 del cuerpo del mensaje
- Metadatos de respuesta

A continuación, se muestra un ejemplo:

```
$ sam remote invoke MySqsQueue --stack-name sqs-example -event hello
```

```
Sending message to SQS queue MySqsQueue
```

```
{
  "MD5OfMessageBody": "5d41402abc4b2a76b9719d911017c592",
  "MessageId": "05c7af65-9ae8-4014-ae28-809d6d8ec652"
}%
```

Cómo enviar un mensaje

1. Proporciona un valor de ID de recurso como argumento para la cola de Amazon SQS. Para obtener información sobre los ID de recursos válidos, consulta el [ID de recurso](#).
2. Indique un evento para que lo envía a la cola de Amazon SQS. Puedes proporcionar el evento en la línea de comandos mediante la opción `--event` o desde un archivo mediante `--event-file`. Si no proporciona un evento, AWS SAM CLI envía un evento vacío.

Uso con Step Functions

Puedes invocar máquinas de estado de Step Functions para iniciar la ejecución. AWS SAM CLI Esperará a que se complete el flujo de trabajo de la máquina de estados y devolverá un resultado del último paso de la ejecución. A continuación, se muestra un ejemplo:

```
$ sam remote invoke HelloWorldStateMachine --stack-name state-machine-example --  
event '{"is_developer": true}'
```

```
Invoking Step Function HelloWorldStateMachine
```

```
"Hello Developer World"%
```

Para invocar una máquina de estado

1. Proporciona un valor de ID de recurso como argumento para la máquina de estado de Step Functions. Para obtener información sobre los ID de recursos válidos, consulta el [ID de recurso](#).
2. Proporciona un evento para enviarlo a su máquina de estado. Puedes proporcionar el evento en la línea de comandos mediante la opción `--event` o desde un archivo mediante `--event-file`. Si no proporciona un evento, AWS SAM CLI envía un evento vacío.

Uso de las opciones del comando `sam remote invoke`

En esta sección se describen algunas de las opciones principales que puede utilizar con el comando `sam remote invoke`. Para obtener una lista completa de opciones, consulta [invocación remota de sam](#).

Paso de un evento al recurso

Utiliza las siguientes opciones para transferir los eventos a sus recursos en la nube:

- `--event`: permite pasar un evento desde la línea de comandos.
- `--event-file`: permite pasar un evento desde un archivo.

Ejemplos de Lambda

Se utiliza `--event` para pasar un evento a la línea de comandos como un valor de cadena:

```
$ sam remote invoke HelloWorldFunction --stack-name sam-app --event '{"message": "hello!"}'
```

Invoking Lambda Function HelloWorldFunction

```
START RequestId: b992292d-1fac-4aa2-922a-c9dc5c6fceab Version: $LATEST
END RequestId: b992292d-1fac-4aa2-922a-c9dc5c6fceab
REPORT RequestId: b992292d-1fac-4aa2-922a-c9dc5c6fceab Duration: 16.41 ms Billed
Duration: 17 ms Memory Size: 128 MB Max Memory Used: 67 MB Init Duration: 185.96
ms
{"statusCode":200,"body":{"\"message\": \"hello!\"}}%
```

Se utiliza `--event-file` para pasar un evento de un archivo y proporcionar la ruta al archivo:

```
$ cat event.json
```

```
{"message": "hello from file"}%
```

```
$ sam remote invoke HelloWorldFunction --stack-name sam-app --event-file event.json
```

Invoking Lambda Function HelloWorldFunction

```
START RequestId: 3bc71f7d-153a-4b1e-8c9a-901d91b1bec9 Version: $LATEST
END RequestId: 3bc71f7d-153a-4b1e-8c9a-901d91b1bec9
REPORT RequestId: 3bc71f7d-153a-4b1e-8c9a-901d91b1bec9 Duration: 21.15 ms Billed
Duration: 22 ms Memory Size: 128 MB Max Memory Used: 67 MB
{"statusCode":200,"body":{"\"message\": \"hello from file\"}}%
```

Pase un evento mediante `stdin`:

```
$ cat event.json
```



```
{"message": "hello from file"}%
```

```
$ cat event.json | sam remote invoke HelloWorldFunction --stack-name sam-app --event-file -
```

Reading event from stdin (you can also pass it from file with --event-file)

Invoking Lambda Function HelloWorldFunction

START RequestId: 85ecc902-8ad0-4a2b-a8c8-9bb4f65f5a7a Version: \$LATEST

END RequestId: 85ecc902-8ad0-4a2b-a8c8-9bb4f65f5a7a

REPORT RequestId: 85ecc902-8ad0-4a2b-a8c8-9bb4f65f5a7a Duration: 1.36 ms Billed

Duration: 2 ms Memory Size: 128 MB Max Memory Used: 67 MB

```
{"statusCode":200,"body":{"\"message\": \"hello from file\"}}%
```

Configura el AWS SAM CLI respuesta de salida

Al invocar un recurso compatible con `sam remote invoke`, la CLI de AWS SAM devuelve una respuesta que contiene lo siguiente:

- Metadatos de la solicitud: metadatos asociados a la solicitud. Esto incluye el identificador de la solicitud y la hora de inicio de la solicitud.
- Respuesta del recurso: la respuesta de su recurso tras ser invocado en la nube.

Puede usar la `--output` opción para configurar la respuesta AWS SAM CLI de salida. Los valores siguientes están disponibles:

- `json`: los metadatos y la respuesta del recurso se devuelven en una estructura JSON. La respuesta contiene el resultado completo del SDK.
- `text`: los metadatos se devuelven en una estructura de texto. La respuesta del recurso se devuelve en el formato de salida del recurso.

A continuación, se muestra un ejemplo de una salida de `json`:

```
$ sam remote invoke --stack-name sam-app --output json
```

Invoking Lambda Function HelloWorldFunction

```
{
  "ResponseMetadata": {
    "RequestId": "3bdf9a30-776d-4a90-94a6-4cccc0fc7b41",
    "HTTPStatusCode": 200,
    "HTTPHeaders": {
      "date": "Mon, 19 Jun 2023 17:15:46 GMT",
      "content-type": "application/json",
      "content-length": "57",
      "connection": "keep-alive",
      "x-amzn-requestid": "3bdf9a30-776d-4a90-94a6-4cccc0fc7b41",
      "x-amzn-remapped-content-length": "0",
      "x-amz-executed-version": "$LATEST",
      "x-amz-log-result":
        "U1RBULQgUmVxdWVzdElkOiAzYmRmOWEzMzZkLTRhOTAtOTRhNi00Y2NjYzBmYzdiNDEgVmVyc2lvbWVjogJExBVEVTV
        "x-amzn-trace-id":
        "root=1-64908d42-17dab270273fcc6b527dd6b8;sampld=0;lineage=2301f8dc:0"
    },
    "RetryAttempts": 0
  },
  "StatusCode": 200,
  "LogResult":
    "U1RBULQgUmVxdWVzdElkOiAzYmRmOWEzMzZkLTRhOTAtOTRhNi00Y2NjYzBmYzdiNDEgVmVyc2lvbWVjogJExBVEVTV
    "ExecutedVersion": "$LATEST",
    "Payload": "{\"statusCode\":200,\"body\":{\"\\\\"message\\\\":\\\\"hello world\\\\"}\\\"}"
}%
```

Al especificar un resultado json, se devuelve la respuesta completa a stdout. A continuación, se muestra un ejemplo:

```
$ sam remote invoke --stack-name sam-app --output json 1> stdout.log
```

```
Invoking Lambda Function HelloWorldFunction
```

```
$ cat stdout.log
```

```
{
  "ResponseMetadata": {
    "RequestId": "d30d280f-8188-4372-bc94-ce0f1603b6bb",
    "HTTPStatusCode": 200,
    "HTTPHeaders": {
      "date": "Mon, 19 Jun 2023 17:35:56 GMT",
```

```

    "content-type": "application/json",
    "content-length": "57",
    "connection": "keep-alive",
    "x-amzn-requestid": "d30d280f-8188-4372-bc94-ce0f1603b6bb",
    "x-amzn-remapped-content-length": "0",
    "x-amz-executed-version": "$LATEST",
    "x-amz-log-result":
"U1RBULQgUmVxdWVzdElk0iBkMzBkMjgwZi04MTg4LTQzNzItYmM5NC1jZTBmMTYwM2I2YmIgVmVyc2lvbjoJExBVEVTV
    "x-amzn-trace-id":
"root=1-649091fc-771473c7778689627a6122b7;sampld=0;lineage=2301f8dc:0"
  },
  "RetryAttempts": 0
},
"StatusCode": 200,
"LogResult":
"U1RBULQgUmVxdWVzdElk0iBkMzBkMjgwZi04MTg4LTQzNzItYmM5NC1jZTBmMTYwM2I2YmIgVmVyc2lvbjoJExBVEVTV
"ExecutedVersion": "$LATEST",
"Payload": "{\"statusCode\":200,\"body\":{\"\\\"message\\\":\\\"hello world\\\"}}\"}
}%

```

A continuación, se muestra un ejemplo de una salida de text:

```
$ sam remote invoke --stack-name sam-app --output text
```

Invoking Lambda Function HelloWorldFunction

```

START RequestId: 4dbacc43-1ec6-47c2-982b-9dc4620144d6 Version: $LATEST
END RequestId: 4dbacc43-1ec6-47c2-982b-9dc4620144d6
REPORT RequestId: 4dbacc43-1ec6-47c2-982b-9dc4620144d6  Duration: 9.13 ms          Billed
Duration: 10 ms  Memory Size: 128 MB      Max Memory Used: 67 MB  Init Duration: 165.50
ms
{"statusCode":200,"body":{"\\\"message\\\":\\\"hello world\\\"}}}%

```

Al especificar un resultado text, se devuelve el resultado del tiempo de ejecución de la función de Lambda (por ejemplo, los registros) a stderr. Se devuelve la carga útil de la función de Lambda a stdout. A continuación, se muestra un ejemplo:

```
$ sam remote invoke --stack-name sam-app --output text 2> stderr.log
```

```

{"statusCode":200,"body":{"\\\"message\\\":\\\"hello world\\\"}}}%

```

```
$ cat stderr.log
```

```
Invoking Lambda Function HelloWorldFunction
```

```
START RequestId: 82273c3b-aa3a-4d16-8f1c-1d2ad3ace891 Version: $LATEST
```

```
END RequestId: 82273c3b-aa3a-4d16-8f1c-1d2ad3ace891
```

```
REPORT RequestId: 82273c3b-aa3a-4d16-8f1c-1d2ad3ace891 Duration: 40.62 ms Billed  
Duration: 41 ms Memory Size: 128 MB Max Memory Used: 68 MB
```

```
$ sam remote invoke --stack-name sam-app --output text 1> stdout.log
```

```
Invoking Lambda Function HelloWorldFunction
```

```
START RequestId: 74acaa9f-5b80-4a5c-b3b8-ffaccb84cbbd Version: $LATEST
```

```
END RequestId: 74acaa9f-5b80-4a5c-b3b8-ffaccb84cbbd
```

```
REPORT RequestId: 74acaa9f-5b80-4a5c-b3b8-ffaccb84cbbd Duration: 2.31 ms Billed  
Duration: 3 ms Memory Size: 128 MB Max Memory Used: 67 MB
```

```
$ cat stdout.log
```

```
{"statusCode":200,"body":{"\message\":"hello world\"},"}%
```

Personalización Boto3 parameters

Puedes usar `sam remote invoke`, la AWS SAM CLI utiliza AWS SDK para Python (Boto3) para interactuar con sus recursos en la nube. Puedes utilizar la opción `--parameter` para personalizar los parámetros Boto3. Para ver una lista de los parámetros admitidos que puede personalizar, consulta [--parameter](#).

Ejemplos

Invoque una función de Lambda para validar los valores de los parámetros y verificar los permisos:

```
$ sam remote invoke HelloWorldFunction --stack-name sam-app --  
parameter InvocationType="DryRun"
```

Utiliza la opción `--parameter` varias veces en un solo comando para proporcionar varios parámetros:

```
$ sam remote invoke HelloWorldFunction --stack-name sam-app --  
parameter InvocationType="Event" --parameter LogType="None"
```

Otras opciones

Para obtener una lista completa de opciones `sam remote invoke`, consulta [invocación remota de sam](#).

Configura el archivo de configuración de su proyecto

Para configurar `sam remote invoke` en su archivo de configuración, utiliza `remote_invoke` en su tabla. El siguiente es un ejemplo de un archivo `samconfig.toml` que configura los valores predeterminados del comando `sam remote invoke`.

```
...  
version =0.1  
  
[default]  
...  
[default.remote_invoke.parameters]  
stack_name = "cloud-app"  
event = '{"message": "Hello!"}'
```

Ejemplos

Para ver un ejemplo básico de uso `sam remote invoke`, consulta [Probar AWS Lambda funciones con AWS SAM control remoto](#) en el blog de AWS informática.

Ejemplos de Kinesis Data Streams

Ejemplos básicos

Envíe un registro de datos a una aplicación de Kinesis Data Streams desde un archivo. La aplicación Kinesis Data Streams se identifica proporcionando un ARN para el ID del recurso:

```
$ sam remote invoke arn:aws:kinesis:us-west-2:01234567890:stream/kinesis-example-  
KinesisStream-BgnLcAey4xUQ --event-file event.json
```

Envíe un evento proporcionado en la línea de comandos a una aplicación de Kinesis Data Streams:

```
$ sam remote invoke KinesisStream --stack-name kinesis-example --event hello-world
```

Putting record to Kinesis data stream KinesisStream

Auto converting value 'hello-world' into JSON '"hello-world"'. If you don't want auto-conversion, please provide

a JSON string as event

```
{
  "ShardId": "shardId-000000000000",
  "SequenceNumber": "49646251411914806775980903986194508740483329854174920706"
}%
```

Obtén el ID físico de la aplicación Kinesis Data Streams. A continuación, proporciona un evento en la línea de comandos:

```
$ sam list resources --stack-name kinesis-example --output json
```

```
[
  {
    "LogicalResourceId": "KinesisStream",
    "PhysicalResourceId": "kinesis-example-KinesisStream-ZgnLcQey4xUQ"
  }
]
```

```
$ sam remote invoke kinesis-example-KinesisStream-ZgnLcQey4xUQ --event hello
```

Putting record to Kinesis data stream KinesisStream

Auto converting value 'hello' into JSON '"hello"'. If you don't want auto-conversion, please provide a JSON string as event

```
{
  "ShardId": "shardId-000000000000",
  "SequenceNumber": "49646251411914806775980904340716841045751814812900261890"
}%
```

Proporciona una cadena JSON en la línea de comandos como un evento:

```
$ sam remote invoke KinesisStream --stack-name kinesis-example --event '{"method":  
"GET", "body": ""'
```

Putting record to Kinesis data stream KinesisStream

```
{
  "ShardId": "shardId-000000000000",
  "SequenceNumber": "49646251411914806775980904492868617924990209230536441858"
}%
```

Envíe un evento vacío a la aplicación Kinesis Data Streams:

```
$ sam remote invoke KinesisStream --stack-name kinesis-example
```

Putting record to Kinesis data stream KinesisStream

```
{
  "ShardId": "shardId-000000000000",
  "SequenceNumber": "49646251411914806775980904866469008589597168190416224258"
}%
```

Devuelve la AWS SAM CLI respuesta en formato JSON:

```
$ sam remote invoke KinesisStream --stack-name kinesis-example --event '{"hello":  
"world"}' --output json
```

Putting record to Kinesis data stream KinesisStream

```
{
  "ShardId": "shardId-000000000000",
  "SequenceNumber": "49646251411914806775980905078409420803696667195489648642",
  "ResponseMetadata": {
    "RequestId": "ebbbd307-3e9f-4431-b67c-f0715e9e353e",
    "HTTPStatusCode": 200,
    "HTTPHeaders": {
      "x-amzn-requestid": "ebbbd307-3e9f-4431-b67c-f0715e9e353e",
      "x-amz-id-2": "Q3yBcgTwtPaQTV26IKclbECmZikUY0zKY+CzcxA84ZHgCkc5T2N/  
ITWg6RP0QcWw8Gn0tNPcEJBEHyVVqboJAPgCritqsvCu",
      "date": "Thu, 09 Nov 2023 18:13:10 GMT",
      "content-type": "application/x-amz-json-1.1",
      "content-length": "110"
    },
    "RetryAttempts": 0
  }
}
```

```
}%
```

Devuelva la salida JSON a stdout:

```
$ sam remote invoke KinesisStream --stack-name kinesis-example --event '{"hello":  
"world"}' --output json 1> stdout.log
```

Putting record to Kinesis data stream KinesisStream

```
$ cat stdout.log  
{  
  "ShardId": "shardId-000000000000",  
  "SequenceNumber": "49646251411914806775980906397777867595039988349006774274",  
  "ResponseMetadata": {  
    "RequestId": "f4290006-d84b-b1cd-a9ee-28306eeb2939",  
    "HTTPStatusCode": 200,  
    "HTTPHeaders": {  
      "x-amzn-requestid": "f4290006-d84b-b1cd-a9ee-28306eeb2939",  
      "x-amz-id-2": "npCqz  
+IBKpoL4sQ1ClbUmxuJlbeA24Fx1UgpIrS6mm2NoIeV2qdZSN5AhNurdssykXajBrXaC9anMhj2eG/h7Hnbf  
+bPuotU",  
      "date": "Thu, 09 Nov 2023 18:33:26 GMT",  
      "content-type": "application/x-amz-json-1.1",  
      "content-length": "110"  
    },  
    "RetryAttempts": 0  
  }  
}%
```

Ejemplos de Lambda

Ejemplos básicos

invoque una función de Lambda al proporcionar el ARN como ID de recurso:

```
$ sam remote invoke arn:aws:lambda:us-west-2:012345678910:function:sam-app-  
HelloWorldFunction-ohRFEn2RuAvp
```

invoque una función de Lambda al proporcionar el ID lógico como ID de recurso:

También debes proporcionar el nombre de la CloudFormation pila mediante la `--stack-name` opción. A continuación, se muestra un ejemplo:


```
$ sam remote invoke HelloWorldFunction --stack-name sam-app
```

Si la aplicación contiene una sola función de Lambda, no es necesario que especifique su ID lógico. Solo puedes proporcionar la opción `--stack-name`. A continuación, se muestra un ejemplo:

```
$ sam remote invoke --stack-name sam-app
```

Para invocar una función de Lambda, proporciona el identificador físico como identificador de recurso:

El ID físico se crea cuando se implementa utilizando CloudFormation.

```
$ sam remote invoke sam-app-HelloWorldFunction-TZvxQRFNv0k4
```

Invoque una función de Lambda de una pila secundaria:

Para este ejemplo, nuestra aplicación contiene la siguiente estructura de directorios:

```
lambda-example
### childstack
#   ### function
#   #   ### __init__.py
#   #   ### app.py
#   #   ### requirements.txt
#   ### template.yaml
### events
#   ### event.json
### samconfig.toml
### template.yaml
```

Para invocar la función de Lambda de nuestra childstack, ejecutamos lo siguiente:

```
$ sam remote invoke ChildStack/HelloWorldFunction --stack-name lambda-example
```

Invoking Lambda Function HelloWorldFunction

START RequestId: 207a864b-e67c-4307-8478-365b004d4bcd Version: \$LATEST

END RequestId: 207a864b-e67c-4307-8478-365b004d4bcd

REPORT RequestId: 207a864b-e67c-4307-8478-365b004d4bcd Duration: 1.27 ms Billed
Duration: 2 ms Memory Size: 128 MB Max Memory Used: 36 MB Init Duration: 111.07
ms

```
{"statusCode": 200, "body": "{\"message\": \"Hello\", \"received_event\": {}}"}%
```

Invoca una función de eventos de Lambda configurada para transmitir respuestas

En este ejemplo, utilizamos la CLI de AWS SAM para inicializar una nueva aplicación sin servidor que contiene una función de Lambda configurada para transmitir su respuesta. Implementamos nuestra aplicación en la nube Nube de AWS y la utilizamos `sam remote invoke` para interactuar con nuestra función en la nube.

Para empezar, es necesario ejecutar el comando `sam init` para crear una nueva aplicación sin servidor. A continuación, se selecciona la plantilla de inicio rápido de Lambda Response Streaming y nombramos la aplicación `lambda-streaming-nodejs-app`.

```
$ sam init
```

```
You can preselect a particular runtime or package type when using the `sam init`  
experience.
```

```
Call `sam init --help` to learn more.
```

```
Which template source would you like to use?
```

```
1 - AWS Quick Start Templates
```

```
2 - Custom Template Location
```

```
Choice: 1
```

```
Choose an AWS Quick Start application template
```

```
1 - Hello World Example
```

```
...
```

```
9 - Lambda Response Streaming
```

```
...
```

```
15 - Machine Learning
```

```
Template: 9
```

```
Which runtime would you like to use?
```

```
1 - go (provided.al2)
```

```
2 - nodejs18.x
```

```
3 - nodejs16.x
```

```
Runtime: 2
```

```
Based on your selections, the only Package type available is Zip.
```

```
We will proceed to selecting the Package type as Zip.
```

```
Based on your selections, the only dependency manager available is npm.
```

```
We will proceed copying the template using npm.
```

Would you like to enable X-Ray tracing on the function(s) in your application? [y/N]: *ENTER*

Would you like to enable monitoring using CloudWatch Application Insights?
For more info, please view <https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/cloudwatch-application-insights.html> [y/N]: *ENTER*

Project name [sam-app]: *lambda-streaming-nodejs-app*

Generating application:

Name: lambda-streaming-nodejs-app
Runtime: nodejs18.x
Architectures: x86_64
Dependency Manager: npm
Application Template: response-streaming
Output Directory: .
Configuration file: lambda-streaming-nodejs-app/samconfig.toml

Next steps can be found in the README file at lambda-streaming-nodejs-app/
README.md

Commands you can use next

=====

[*] Create pipeline: `cd lambda-streaming-nodejs-app && sam pipeline init --bootstrap`
[*] Validate SAM template: `cd lambda-streaming-nodejs-app && sam validate`
[*] Test Function in the Cloud: `cd lambda-streaming-nodejs-app && sam sync --stack-name {stack-name} --watch`

AWS SAMCLICrea nuestro proyecto con la siguiente estructura:

```
lambda-streaming-nodejs-app
### README.md
### __tests__
#   ### unit
#       ### index.test.js
### package.json
### samconfig.toml
### src
#   ### index.js
```

```
### template.yaml
```

El siguiente es un ejemplo de nuestro código de la función de Lambda:

```
exports.handler = awslambda.streamifyResponse(
  async (event, responseStream, context) => {
    const httpResponseMetadata = {
      statusCode: 200,
      headers: {
        "Content-Type": "text/html",
        "X-Custom-Header": "Example-Custom-Header"
      }
    };

    responseStream = awslambda.HttpResponseStream.from(responseStream,
      httpResponseMetadata);
    // It's recommended to use a `pipeline` over the `write` method for more complex
    use cases.
    // Learn more: https://docs.aws.amazon.com/lambda/latest/dg/configuration-
    response-streaming.html
    responseStream.write("<html>");
    responseStream.write("<p>First write!</p>");

    responseStream.write("<h1>Streaming h1</h1>");
    await new Promise(r => setTimeout(r, 1000));
    responseStream.write("<h2>Streaming h2</h2>");
    await new Promise(r => setTimeout(r, 1000));
    responseStream.write("<h3>Streaming h3</h3>");
    await new Promise(r => setTimeout(r, 1000));

    // Long strings will be streamed
    const loremIpsum1 = "Lorem ipsum dolor sit amet, consectetur adipiscing elit.
    Quisque vitae mi tincidunt tellus ultricies dignissim id et diam. Morbi pharetra eu
    nisi et finibus. Vivamus diam nulla, vulputate et nisl cursus, pellentesque vehicula
    libero. Cras imperdiet lorem ante, non posuere dolor sollicitudin a. Vestibulum ipsum
    lacus, blandit nec augue id, lobortis dictum urna. Vestibulum ante ipsum primis in
    faucibus orci luctus et ultrices posuere cubilia curae; Morbi auctor orci eget tellus
    aliquam, non maximus massa porta. In diam ante, pulvinar aliquam nisl non, elementum
    hendrerit sapien. Vestibulum massa nunc, mattis non congue vitae, placerat in quam.
    Nam vulputate lectus metus, et dignissim erat varius a.";
    responseStream.write(`<p>${loremIpsum1}</p>`);
    await new Promise(r => setTimeout(r, 1000));
```

```
    responseStream.write("<p>DONE!</p>");
    responseStream.write("</html>");
    responseStream.end();
  }
};
```

A continuación, se presenta un ejemplo del archivo `template.yaml`. La transmisión de respuestas para nuestra función de Lambda se configura mediante la propiedad `FunctionUrlConfig`.

```
AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31

Description: >
  Sample SAM Template for lambda-streaming-nodejs-app

Resources:
  StreamingFunction:
    Type: AWS::Serverless::Function
    Properties:
      CodeUri: src/
      Handler: index.handler
      Runtime: nodejs18.x
      Architectures:
        - x86_64
      Timeout: 10
      FunctionUrlConfig:
        AuthType: AWS_IAM
        InvokeMode: RESPONSE_STREAM

Outputs:
  StreamingFunction:
    Description: "Streaming Lambda Function ARN"
    Value: !GetAtt StreamingFunction.Arn
  StreamingFunctionURL:
    Description: "Streaming Lambda Function URL"
    Value: !GetAtt StreamingFunctionUrl.FunctionUrl
```

Por lo general, puede usar `sam build` y `sam deploy --guided` para crear e implementar una aplicación de producción. En este ejemplo, accederemos a un entorno de desarrollo y utilizaremos el comando `sam sync` para crear e implementar nuestra aplicación.

 Note

Se recomienda usar el comando `sam sync` para entornos de desarrollo. Para obtener más información, consulta [Introducción al uso sam sync sincronizar con Nube de AWS](#).

Antes de ejecutar `sam sync`, debe verificar que el proyecto esté configurado correctamente en el archivo `samconfig.toml`. Asimismo, recuerde verificar los valores de `stack_name` y `watch`. Con estos valores especificados en el archivo de configuración, no es necesario proporcionarlos en la línea de comandos.

```
version = 0.1

[default]
[default.global.parameters]
stack_name = "lambda-streaming-nodejs-app"

[default.build.parameters]
cached = true
parallel = true

[default.validate.parameters]
lint = true

[default.deploy.parameters]
capabilities = "CAPABILITY_IAM"
confirm_changeset = true
resolve_s3 = true
s3_prefix = "lambda-streaming-nodejs-app"
region = "us-west-2"
image_repositories = []

[default.package.parameters]
resolve_s3 = true

[default.sync.parameters]
watch = true

[default.local_start_api.parameters]
warm_containers = "EAGER"

[default.local_start_lambda.parameters]
```

```
warm_containers = "EAGER"
```

A continuación, ejecuta `sam sync` para crear e implementar nuestra aplicación. Como la opción `--watch` está configurada en nuestro archivo de configuración, la CLI de AWS SAM compilará la aplicación, la implementará y estará pendiente de los cambios.

```
$ sam sync
```

```
The SAM CLI will use the AWS Lambda, Amazon API Gateway, and AWS StepFunctions APIs to
upload your code
without
```

```
performing a CloudFormation deployment. This will cause drift in your CloudFormation
stack.
```

```
**The sync command should only be used against a development stack**.
```

```
Queued infra sync. Waiting for in progress code syncs to complete...
```

```
Starting infra sync.
```

```
Building codeuri:
```

```
/Users/.../lambda-streaming-nodejs-app/src runtime: nodejs18.x metadata: {}
architecture: x86_64 functions: StreamingFunction
package.json file not found. Continuing the build without dependencies.
```

```
Running NodejsNpmBuilder:CopySource
```

```
Build Succeeded
```

```
Successfully packaged artifacts and wrote output template to file /var/
folders/45/5ct135bx3fn2551_ptl5g6_80000gr/T/tmpavrzdhgp.
```

```
Execute the following command to deploy the packaged template
```

```
sam deploy --template-file /var/folders/45/5ct135bx3fn2551_ptl5g6_80000gr/T/
tmpavrzdhgp --stack-name <YOUR STACK NAME>
```

```
Deploying with following values
```

```
=====
```

```
Stack name                : lambda-streaming-nodejs-app
```

```

    Region                : us-west-2
    Disable rollback      : False
    Deployment s3 bucket  : aws-sam-cli-managed-default-samclisam-s3-demo-
bucket-1a4x26zbcdkqr
    Capabilities          : ["CAPABILITY_NAMED_IAM",
"CAPABILITY_AUTO_EXPAND"]
    Parameter overrides   : {}
    Signing Profiles      : null

```

Initiating deployment

=====

2023-06-20 12:11:16 - Waiting for stack create/update to complete

CloudFormation events from stack operations (refresh every 0.5 seconds)

ResourceStatus	ResourceType	LogicalResourceId	ResourceStatusReason
CREATE_IN_PROGRESS	AWS::CloudFormation::Stack	lambda-streaming-	
	ack	nodejs-app	
succeeded			
CREATE_IN_PROGRESS	AWS::IAM::Role	StreamingFunctionRole	-
CREATE_IN_PROGRESS	AWS::CloudFormation::Stack	AwsSamAutoDependencyLay	-
	ack	erNestedStack	
CREATE_IN_PROGRESS	AWS::IAM::Role	StreamingFunctionRole	Resource
Initiated			
CREATE_IN_PROGRESS	AWS::CloudFormation::Stack	AwsSamAutoDependencyLay	Resource
creation	ack	erNestedStack	
Initiated			
CREATE_COMPLETE	AWS::IAM::Role	StreamingFunctionRole	-
CREATE_COMPLETE	AWS::CloudFormation::Stack	AwsSamAutoDependencyLay	-
	ack	erNestedStack	

CREATE_IN_PROGRESS	AWS::Lambda::Function	StreamingFunction	-
CREATE_IN_PROGRESS creation	AWS::Lambda::Function	StreamingFunction	Resource
Initiated			
CREATE_COMPLETE	AWS::Lambda::Function	StreamingFunction	-
CREATE_IN_PROGRESS	AWS::Lambda::Url	StreamingFunctionUrl	-
CREATE_IN_PROGRESS creation	AWS::Lambda::Url	StreamingFunctionUrl	Resource
Initiated			
CREATE_COMPLETE	AWS::Lambda::Url	StreamingFunctionUrl	-
CREATE_COMPLETE	AWS::CloudFormation::St	lambda-streaming-	-
	ack	nodejs-app	

CloudFormation outputs from deployed stack			

Outputs			

Key	StreamingFunction		
Description	Streaming Lambda Function ARN		
Value	arn:aws:lambda:us-west-2:012345678910:function:lambda-streaming-		
nodejs-app-			
StreamingFunction-gUmh0833A0vZ			
Key	StreamingFunctionURL		
Description	Streaming Lambda Function URL		
Value	https://wxgkcc2dyntgtrwhf2dgdcvylu0rnnof.lambda-url.us-		
west-2.on.aws/			

```
Stack creation succeeded. Sync infra completed.
```

```
Infra sync completed.
```

Ahora que la función está implementada en la nube, puede utilizar `sam remote invoke` para interactuar con ella. La CLI de AWS SAM detecta automáticamente que la función está configurada para la transmisión de respuestas e inmediatamente comienza a generar una respuesta transmitida de la función en tiempo real.

```
$ sam remote invoke StreamingFunction
```

```
Invoking Lambda Function StreamingFunction
```

```
{"statusCode":200,"headers":{"Content-Type":"text/html","X-Custom-Header":"Example-Custom-Header"}}<html><p>First write!</p><h1>Streaming h1</h1><h2>Streaming h2</h2><h3>Streaming h3</h3><p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque vitae mi tincidunt tellus ultricies dignissim id et diam. Morbi pharetra eu nisi et finibus. Vivamus diam nulla, vulputate et nisl cursus, pellentesque vehicula libero. Cras imperdiet lorem ante, non posuere dolor sollicitudin a. Vestibulum ipsum lacus, blandit nec augue id, lobortis dictum urna. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia curae; Morbi auctor orci eget tellus aliquam, non maximus massa porta. In diam ante, pulvinar aliquam nisl non, elementum hendrerit sapien. Vestibulum massa nunc, mattis non congue vitae, placerat in quam. Nam vulputate lectus metus, et dignissim erat varius a.</p><p>DONE!</p></html>START RequestId: 1e4cdf04-60de-4769-b3a2-c1481982deb4 Version: $LATEST
END RequestId: 1e4cdf04-60de-4769-b3a2-c1481982deb4
REPORT RequestId: 1e4cdf04-60de-4769-b3a2-c1481982deb4  Duration: 4088.66 ms
Billed Duration: 4089 ms          Memory Size: 128 MB    Max Memory Used: 68 MB   Init
Duration: 168.45 ms
```

Cuando modificamos el código de la función, la CLI de AWS SAM detecta los cambios realizados al instante y los implementa de inmediato. Este es un ejemplo del resultado de la CLI de AWS SAM después de realizar cambios en el código de nuestra función:

```
Syncing Lambda Function StreamingFunction...
```

```
Building codeuri:
```

```

/Users/.../lambda-streaming-nodejs-app/src runtime: nodejs18.x metadata: {}
architecture:
x86_64 functions: StreamingFunction

package.json file not found. Continuing the build without dependencies.

Running NodejsNpmBuilder:CopySource

Finished syncing Lambda Function StreamingFunction.

Syncing Layer StreamingFunctione9cfe924DepLayer...

SyncFlow [Layer StreamingFunctione9cfe924DepLayer]: Skipping resource update as the
content didn't change

Finished syncing Layer StreamingFunctione9cfe924DepLayer.

```

Ahora puede volver a utilizar `sam remote invoke` para interactuar con la función en la nube y probar los cambios.

Ejemplos de SQS

Ejemplos básicos

Invocar una cola de Amazon SQS al proporcionar el ARN como ID de recurso:

```

$ sam remote invoke arn:aws:sqs:us-west-2:01234567890:sqs-example-4DonhBsjsW1b --
event '{"hello": "world"}' --output json

```

Sending message to SQS queue MySqsQueue

```

{
  "MD5OfMessageBody": "49dfdd54b01cbcd2d2ab5e9e5ee6b9b9",
  "MessageId": "4f464cdd-15ef-4b57-bd72-3ad225d80adc",
  "ResponseMetadata": {
    "RequestId": "95d39377-8323-5ef0-9223-ceb198bd09bd",
    "HTTPStatusCode": 200,
    "HTTPHeaders": {
      "x-amzn-requestid": "95d39377-8323-5ef0-9223-ceb198bd09bd",
      "date": "Wed, 08 Nov 2023 23:27:26 GMT",
      "content-type": "application/x-amz-json-1.0",
      "content-length": "106",

```

```

    "connection": "keep-alive"
  },
  "RetryAttempts": 0
}
}%

```

Ejemplos de Step Functions

Ejemplos básicos

Para invocar una máquina de estados, proporciona su ID físico como ID de recurso:

En primer lugar, usa `sam list resources` para obtener el ID físico:

```

$ sam list resources --stack-name state-machine-example --output json

[
  {
    "LogicalResourceId": "HelloWorldStateMachine",
    "PhysicalResourceId": "arn:aws:states:us-
west-2:513423067560:stateMachine:HelloWorldStateMachine-z69tFEUx0F66"
  },
  {
    "LogicalResourceId": "HelloWorldStateMachineRole",
    "PhysicalResourceId": "simple-state-machine-HelloWorldStateMachineRole-
PduA0BDGuFXw"
  }
]

```

A continuación, invoca la máquina de estados mediante el ID físico como ID de recurso. Transmite un evento en la línea de comandos con la opción `--event`:

```

$ sam remote invoke arn:aws:states:us-
west-2:01234567890:stateMachine:HelloWorldStateMachine-z69tFEUx0F66 --
event '{"is_developer": true}'

```

```

Invoking Step Function arn:aws:states:us-
west-2:01234567890:stateMachine:HelloWorldStateMachine-z69tFEUx0F66
"Hello Developer World"%

```

Invoque una máquina de estados pasando un evento vacío:

```

$ sam remote invoke HelloWorldStateMachine --stack-name state-machine-example

```

```
Invoking Step Function HelloWorldStateMachine
```

```
"Hello World"%
```

Vínculos relacionados

Para obtener documentación relacionada `sam remote invoke` con el uso del AWS SAMCLI, consulte lo siguiente:

- [invocación remota de sam](#)
- [AWS SAM CLI solución de problemas](#)

Automatice las pruebas de integración local con AWS SAM

Si bien puede usarlo [Introducción a las pruebas con invocar sam local](#) para probar el código manualmente, AWS SAM también le permite probar su código mediante pruebas de integración automatizadas. Las pruebas de integración le ayudan a detectar problemas en las primeras etapas del ciclo de desarrollo, a mejorar la calidad del código, a ahorrar tiempo y, al mismo tiempo, a reducir los costos.

Para crear pruebas de integración automatizadas AWS SAM, primero debe ejecutar pruebas en las funciones de eventos de Lambda locales antes de implementarlas en la AWS nube. El comando [Introducción a las pruebas con sam local start-lambda](#) inicia un punto de conexión local que emula el punto de conexión de invocación de Lambda. Puedes invocarlo desde sus pruebas automatizadas. Debido a que este punto de conexión emula el punto de conexión de invocación de Lambda, puede escribir pruebas una vez y ejecutarlas (sin ninguna modificación) en la función de Lambda local o en una función de Lambda implementada. También puede ejecutar las mismas pruebas en una AWS SAM pila implementada en su CI/CD proceso.

Así es como funciona el proceso:

1. Inicie el punto de conexión Lambda local.

Inicie el punto final Lambda local ejecutando el siguiente comando en el directorio que contiene la plantilla AWS SAM :

```
sam local start-lambda
```

Este comando inicia un punto de conexión local en `http://127.0.0.1:3001` que emula AWS Lambda. Puede ejecutar las pruebas automatizadas en este punto de conexión Lambda local. Al invocar este punto final mediante el SDK AWS CLI o el SDK, ejecuta localmente la función Lambda que se especifica en la solicitud y devuelve una respuesta.

2. Ejecute una prueba de integración en el punto de conexión Lambda local.

En la prueba de integración, puede usar el AWS SDK para invocar la función Lambda con los datos de la prueba, esperar la respuesta y comprobar que la respuesta es la esperada. Para ejecutar la prueba de integración localmente, debe configurar el SDK de AWS para que envíe la llamada a la API invocación de Lambda al punto de conexión local de Lambda iniciado en el paso anterior.

El siguiente es un ejemplo de Python (los AWS SDK para otros lenguajes tienen configuraciones similares):

```
import boto3
import botocore

# Set "running_locally" flag if you are running the integration test locally
running_locally = True

if running_locally:

    # Create Lambda SDK client to connect to appropriate Lambda endpoint
    lambda_client = boto3.client('lambda',
        region_name="us-west-2",
        endpoint_url="http://127.0.0.1:3001",
        use_ssl=False,
        verify=False,
        config=botocore.client.Config(
            signature_version=botocore.UNSIGNED,
            read_timeout=15,
            retries={'max_attempts': 0},
        )
    )
else:
    lambda_client = boto3.client('lambda')

# Invoke your Lambda function as you normally usually do. The function will run
# locally if it is configured to do so
```

```
response = lambda_client.invoke(FunctionName="HelloWorldFunction")

# Verify the response
assert response == "Hello World"
```

Puede usar este código para probar las funciones de Lambda implementadas configurando `running_locally` en `False`. Esto configura el AWS SDK al que se debe conectar AWS Lambda en la AWS nube.

Genere cargas útiles de eventos de muestra con AWS SAM

Para probar sus funciones de eventos de Lambda, puede generar y personalizar ejemplos de cargas útiles de eventos que imiten los datos que recibirán sus funciones de Lambda cuando sean activadas por otros servicios. AWS Esto incluye servicios de como API Gateway, CloudFormation o Amazon S3, entre otros.

La generación de cargas útiles de eventos de muestra le ayuda a probar el comportamiento de la función de Lambda con una variedad de entradas diferentes sin necesidad de trabajar en un entorno en vivo. Este enfoque también ahorra tiempo en comparación con la creación manual de muestras de eventos de AWS servicio para probar las funciones.

Para ver la lista completa de servicios para los que puede generar cargas útiles de eventos de muestra, utiliza este comando:

```
sam local generate-event --help
```

Para ver la lista de opciones que puede usar para un servicio concreto, utiliza este comando:

```
sam local generate-event [SERVICE] --help
```

Ejemplos:

```
#Generates the event from S3 when a new object is created
sam local generate-event s3 put

# Generates the event from S3 when an object is deleted
sam local generate-event s3 delete
```

Prueba y depuración de funciones duraderas

Probar y depurar funciones duraderas a nivel local funciona de forma similar a las funciones Lambda normales, con soporte automático para puntos de control y reproducción. Esta guía cubre los escenarios de prueba y las técnicas de solución de problemas más comunes.

Flujo de trabajo de pruebas locales

Cuando se prueban funciones duraderas a nivel local, el flujo de trabajo es diferente al de las funciones Lambda normales:

Flujo de trabajo de pruebas de funciones duradero

1. Cree su aplicación:

```
$ sam build
```

2. Invoque su función duradera:

```
$ sam local invoke MyDurableFunction --durable-execution-name test
```

3. Compruebe el historial de ejecución si es necesario:

```
$ sam local execution history execution-id
```

Escenarios de prueba comunes

Probar el comportamiento de los puntos de control

Para comprobar que su funcionamiento es correcto, los puntos de control indican:

```
# Example Python durable function
def handler(event, context):
    # This will create a checkpoint
    context.wait(10) # Wait 10 seconds

    # Function resumes here after wait
    return {"message": "Function resumed after wait"}
```

Al invocar esta función de forma local, el período de espera se gestiona automáticamente.

Probando escenarios de devolución de llamadas

Para las funciones que esperan llamadas externas:

1. Inicie su función duradera que espera una devolución de llamada
2. En otro terminal, resuelve la devolución de llamada:

```
$ sam local callback succeed callback-id
```

3. Observe cómo se reanuda la ejecución de la función

Resolución de problemas

La función duradera no se ejecuta correctamente

Problema: la función no se comporta como una función duradera.

Soluciones:

- Comprueba que `DurableConfig` esté configurado en tu plantilla SAM
- Asegúrese de que el código de su función utilice métodos de SDK de funciones duraderas (por ejemplo, `context.wait()`)
- Comprueba que estás utilizando un entorno de ejecución compatible (TypeScript, JavaScript, Python)

No se puede recuperar el historial de ejecución

Problema: el `local execution history` comando no devuelve ningún resultado.

Soluciones:

- Compruebe que el identificador de ejecución es correcto
- Compruebe que la función se haya invocado al menos una vez

Los comandos callback no funcionan

Problema: los comandos de devolución de llamada no resuelven las operaciones pendientes.

Soluciones:

- Compruebe que el identificador de devolución de llamada sea correcto
- Asegúrese de que la función esté realmente esperando una devolución de llamada
- Compruebe que está utilizando la sintaxis correcta del comando de devolución de llamada

Consejos de depuración

- Utilice el historial de ejecución: revise el historial de ejecución para comprender el flujo de su función duradera
- Realice pruebas incrementales: comience con operaciones de espera sencillas antes de añadir una lógica compleja
- Utilice el registro detallado: habilite el registro detallado para rastrear el flujo de ejecución

Más información

Para obtener más información sobre las pruebas y la depuración, consulte:

- [Introducción a las pruebas con invocar sam local](#)- Documentación de invocación local
- [sam local execution history](#)- Historial de ejecuciones

Depure su aplicación sin servidor con AWS SAM

Después de probar la aplicación, ya puede depurar cualquier problema que haya encontrado. Con la interfaz de línea de AWS SAM comandos (CLI), puede probar y depurar localmente su aplicación sin servidor antes de subirla a la nube. AWS La depuración de la aplicación identifica y corrige los problemas o errores de la aplicación.

Puede utilizarla AWS SAM para realizar una depuración gradual, que consiste en ejecutar el código una línea o una instrucción a la vez. Cuando invoca localmente una función de eventos de Lambda en modo de depuración dentro AWS SAMCLI del, puede adjuntarle un depurador. Con el depurador, puede revisar el código línea por línea, ver los valores de varias variables y solucionar problemas del mismo modo que lo haría con cualquier otra aplicación. Puedes comprobar si tu aplicación se comporta como se espera, depurar los errores y solucionar cualquier problema antes de seguir los pasos de empaquetado e implementación de la aplicación.

Note

Si la aplicación incluye una o más capas, al ejecutar y depurar la aplicación de forma local, el paquete de capas se descarga y se almacena en caché en el host local. Para obtener más información, consulta [Cómo se almacenan las capas en caché local](#).

Temas

- [Depure funciones localmente con AWS SAM](#)
- [Pase varios argumentos de tiempo de ejecución al depurar con AWS SAM](#)
- [Valide sus AWS SAM aplicaciones con CloudFormation Linter](#)

Depure funciones localmente con AWS SAM

Puede utilizarlas AWS SAM con una variedad de AWS kits de herramientas y depuradores para probar y depurar sus aplicaciones sin servidor de forma local. Step-through la depuración de las funciones de eventos de Lambda le permite identificar y corregir los problemas de la aplicación, línea por línea o instrucción, en su entorno local.

Algunas de las formas en que puede realizar una depuración gradual local son estableciendo puntos de interrupción, inspeccionando variables y ejecutando el código de la función línea por línea.

La depuración gradual local estrecha el ciclo de retroalimentación, ya que te permite encontrar y solucionar los problemas que puedan surgir en la nube.

Puede utilizar los AWS kits de herramientas para realizar la depuración y también puede AWS SAM ejecutarla en modo de depuración. Para obtener más información, consulte los temas de esta sección.

Utilización AWS Kits de herramientas

AWS Los kits de herramientas son complementos de entornos de desarrollo (IDE) integrados que permiten realizar muchas tareas de depuración habituales, como establecer puntos de interrupción, inspeccionar variables y ejecutar el código de la función línea por línea. AWS Los kits de herramientas facilitan el desarrollo, la depuración y la implementación de aplicaciones sin servidor creadas con ellas. AWS SAM Proporcionan una experiencia para crear, probar, depurar, implementar e invocar funciones de Lambda integradas en su IDE.

Para obtener más información sobre los AWS kits de herramientas con los que puede utilizar AWS SAM, consulte lo siguiente:

- [AWS Toolkit for Visual Studio Code](#)
- [AWS Cloud9](#)
- [AWS Toolkit for JetBrains](#)

Hay una variedad de AWS kits de herramientas que funcionan con diferentes combinaciones de IDE y tiempos de ejecución. En la siguiente tabla se enumeran las IDE/runtime combinaciones más comunes que permiten la depuración gradual de las aplicaciones: AWS SAM

IDE	Tiempo de ejecución	AWS Kit de herramientas	Instrucciones para la depuración gradual
Visual Studio Code	<ul style="list-style-type: none">• Node.js• Python• .NET• Java• Go	AWS Toolkit for Visual Studio Code	Trabajar con Aplicación sin servidor de AWS en la Guía del usuario de AWS Toolkit for Visual Studio Code

IDE	Tiempo de ejecución	AWS Kit de herramientas	Instrucciones para la depuración gradual
AWS Cloud9	<ul style="list-style-type: none"> Node.js Python 	AWS Cloud9 ¹ con el AWS kit de herramientas activado	Trabajar con aplicaciones AWS sin servidor mediante el AWS kit de herramientas de la Guía del AWS Cloud9 usuario.
WebStorm	Node.js	AWS Toolkit for JetBrains ²	Ejecución (invocación) o depuración de una función local en el AWS Toolkit for JetBrains
PyCharm	Python	AWS Toolkit for JetBrains ²	Ejecución (invocación) o depuración de una función local en el AWS Toolkit for JetBrains
Rider	.NET	AWS Toolkit for JetBrains ²	Ejecución (invocación) o depuración de una función local en el AWS Toolkit for JetBrains
IntelliJ	Java	AWS Toolkit for JetBrains ²	Ejecución (invocación) o depuración de una función local en el AWS Toolkit for JetBrains

IDE	Tiempo de ejecución	AWS Kit de herramientas	Instrucciones para la depuración gradual
GoLand	Go	AWS Toolkit for JetBrains ²	Ejecución (invocación) o depuración de una función local en el AWS Toolkit for JetBrains

Notas:

1. Para poder AWS Cloud9 depurar AWS SAM aplicaciones paso a paso, el AWS kit de herramientas debe estar activado. Para obtener más información, consulte [Habilitar el AWS kit de herramientas en la Guía del](#) usuario.AWS Cloud9
2. Para utilizar las AWS Toolkit for JetBrains AWS SAM aplicaciones de depuración paso a paso, primero debe instalarlas y configurarlas siguiendo las instrucciones que se encuentran en [Instalación del AWS Toolkit for JetBrains](#). AWS Toolkit for JetBrains

Ejecutar AWS SAM localmente en modo de depuración

[Además de integrarse con los AWS kits de herramientas, también puedes ejecutarlo AWS SAM en modo de depuración para adjuntarlo a depuradores de terceros, como ptvsd o delve.](#)

Para ejecutar AWS SAM en modo de depuración, usa comandos o con la opción o. [invocar sam local sam local start-api](#) --debug-port -d

Por ejemplo:

```
# Invoke a function locally in debug mode on port 5858
sam local invoke -d 5858 <function logical id>

# Start local API Gateway in debug mode on port 5858
sam local start-api -d 5858
```

Note

Si utiliza `sam local start-api`, la instancia de la API Gateway local expone todas las funciones de Lambda;. Sin embargo, debido a que solo se puede especificar un puerto de

depuración, las funciones solo se pueden depurar de una en una. Debe llamar a su API antes de que AWS SAMCLI se vincule al puerto, lo que permite que el depurador se conecte.

Pase varios argumentos de tiempo de ejecución al depurar con AWS SAM

Puede optar por pasar argumentos de tiempo de ejecución adicionales AWS SAM para inspeccionar los problemas y solucionar las variables de forma más eficaz. De este modo, se proporciona un mayor control y flexibilidad para el proceso de depuración, lo que puede ayudarle a personalizar las configuraciones y los entornos de ejecución.

Para pasar argumentos de tiempo de ejecución adicionales al depurar la función, usa la variable de entorno `DEBUGGER_ARGS`. Esto pasa una cadena de argumentos directamente al comando de ejecución que AWS SAMCLI utiliza para iniciar la función.

Por ejemplo, si quieres cargar un depurador como i KPdb en el tiempo de ejecución de tu función de Python, puedes pasar lo siguiente como `DEBUGGER_ARGS`: `-m ikpdb --ikpdb-port=5858 --ikpdb-working-directory=/var/task/ --ikpdb-client-working-directory=/myApp --ikpdb-address=0.0.0.0`. Esto cargaría i KPdb en tiempo de ejecución con los demás argumentos que hayas especificado.

En este caso, el comando AWS SAMCLI completo sería:

```
DEBUGGER_ARGS="-m ikpdb --ikpdb-port=5858 --ikpdb-working-directory=/var/task/ --ikpdb-client-working-directory=/myApp --ikpdb-address=0.0.0.0" echo {} | sam local invoke -d 5858 myFunction
```

Puede pasar los argumentos del depurador a las funciones de todos los tiempos de ejecución.

Valide sus AWS SAM aplicaciones con CloudFormation Linter

AWS CloudFormation Linter (cfn-lint) es una herramienta de código abierto que puede utilizar para realizar una validación detallada de sus plantillas. CloudFormation CFN-lint contiene reglas que se guían por la especificación del recurso. CloudFormation Use cfn-lint para comparar sus recursos con esas reglas y recibir mensajes detallados sobre errores, advertencias o sugerencias informativas. También puede crear sus propias reglas personalizadas para validarlas. [Para obtener más información sobre cfn-lint, consulte cfn-lint en el repositorio.CloudFormation GitHub](#)

Puedes usar `cfn-lint` para validar tus plantillas () a través de la interfaz de línea de AWS SAM comandos AWS Serverless Application Model (AWS SAM) ejecutando esta opción. `aws-sam-cli validate --lint`

```
sam validate --lint
```

Para personalizar el comportamiento de `cfn-lint`, como crear reglas personalizadas o especificar opciones de validación, puede definir un archivo de configuración. Para obtener más información, consulte el [archivo Config](#) en el repositorio `cfn-lint` CloudFormation GitHub . Cuando ejecute `sam validate --lint`, se aplicará el comportamiento de `cfn-lint` definido en el archivo de configuración.

Ejemplos

Realice la validación de `cfn-lint` en una plantilla AWS SAM

```
sam validate --lint --template myTemplate.yaml
```

Más información

Para obtener más información acerca del comando `sam validate`, consulte [sam validate](#).

Implemente su aplicación y sus recursos con AWS SAM

La implementación de su aplicación aprovisiona y configura sus AWS recursos en la AWS nube, lo que hace que su aplicación se ejecute en la nube. AWS SAM utiliza [CloudFormation](#) como mecanismo de despliegue subyacente. AWS SAM utiliza los artefactos de compilación que se crean al ejecutar el `sam build` comando como entradas estándar para implementar la aplicación sin servidor.

Con AWS SAM ella, puede implementar su aplicación sin servidor manualmente o puede automatizar las implementaciones. Para automatizar las implementaciones, utilice AWS SAM canalizaciones con el sistema de integración e implementación continuas (CI/CD) de su elección. La canalización de implementación es una secuencia automatizada de pasos que se llevan a cabo para lanzar una nueva versión de la aplicación sin servidor.

En los temas de esta sección se proporcionan instrucciones sobre las implementaciones automatizadas y manuales. Para implementar la aplicación manualmente, utilizará comandos de la CLI de AWS SAM. Para automatizar las implementaciones, consulte los temas de esta sección. Proporcionan específicamente contenido detallado sobre la automatización de las implementaciones mediante canalizaciones y un sistema. CI/CD Esto incluye la generación de una canalización inicial, la configuración de la automatización, la solución de problemas de las implementaciones, el uso de la autenticación de usuarios de OpenID Connect (OIDC) y la carga de archivos locales en el momento de la implementación.

Temas

- [Introducción a la implementación con AWS SAM](#)
- [Opciones para implementar su aplicación con AWS SAM](#)
- [Uso CI/CD de sistemas y canalizaciones para implementar con AWS SAM](#)
- [Introducción al uso `sam sync` sincronizar con Nube de AWS](#)

Introducción a la implementación con AWS SAM

Utilice el AWS Serverless Application Model comando Command Line Interface (AWS SAMCLI) `sam deploy` para implementar la Nube de AWS aplicación sin servidor en.

- Para obtener una introducción al AWS SAMCLI, consulte [¿Qué es el AWS SAM CLI?](#).
- Para ver una lista de opciones de comandos `sam deploy`, consulta [sam deploy](#).

- Para ver un ejemplo de uso de `sam deploy` durante un flujo de trabajo de desarrollo típico, consulta [Paso 3: Implemente la aplicación en Nube de AWS](#).

Temas

- [Requisitos previos](#)
- [Implementación de aplicaciones mediante `sam deploy`](#)
- [Prácticas recomendadas](#)
- [Opciones para una implementación de `sam`](#)
- [Resolución de problemas](#)
- [Ejemplos](#)
- [Más información](#)

Requisitos previos

Para usar `sam deploy`, completa lo siguiente para instalar la CLI de AWS SAM:

- [AWS SAM requisitos previos](#).
- [Instala la AWS SAM CLI](#).

Antes de usar `sam deploy`, se recomienda tener conocimientos básicos de lo siguiente:

- [Configuración de la AWS SAM de CLI](#).
- [Crea tu aplicación en AWS SAM](#).
- [Introducción a la construcción con AWS SAM](#).

Implementación de aplicaciones mediante `sam deploy`

La primera vez que implementa una aplicación sin servidor, utiliza la opción `--guided`. La CLI de AWS SAM lo guiará a través de un flujo interactivo para configurar los ajustes de implementación de la aplicación.

Para implementar una aplicación mediante el flujo interactivo

1. Vaya al directorio raíz del proyecto. Esta es la misma ubicación que la AWS SAM plantilla.

```
$ cd sam-app
```

2. Use el siguiente comando:

```
$ sam deploy --guided
```

3. Durante el flujo interactivo, la CLI de AWS SAM le indica las opciones para configurar los ajustes de implementación de la aplicación.

Los corchetes ([]) indican los valores predeterminados. Deje la respuesta en blanco para seleccionar el valor predeterminado. Los valores predeterminados se obtienen de los siguientes archivos de configuración:

- `~/.aws/config`— La configuración general AWS de tu cuenta.
- `~/.aws/credentials`— Las credenciales AWS de su cuenta.
- `<project>/samconfig.toml`: el archivo de configuración de su proyecto.

Proporciona valores respondiendo a las instrucciones de la CLI de AWS SAM. Por ejemplo, puede introducir **y** para sí, **n** para no o cadenas de valores.

La CLI de AWS SAM escriba las respuestas en el archivo `samconfig.toml` del proyecto. Para implementaciones posteriores, puede utilizar la implementación `sam deploy` con estos valores configurados. Para volver a configurar estos valores, utiliza `sam deploy --guided` o modifique directamente los archivos de configuración.

El siguiente es un ejemplo de output:

```
sam-app $ sam deploy --guided

Configuring SAM deploy
=====

    Looking for config file [samconfig.toml] : Found
    Reading default arguments : Success

    Setting default arguments for 'sam deploy'
    =====
    Stack Name [sam-app]: ENTER
    AWS Region [us-west-2]: ENTER
```

```

#Shows you resources changes to be deployed and require a 'Y' to initiate
deploy
  Confirm changes before deploy [Y/n]: ENTER
  #SAM needs permission to be able to create roles to connect to the
resources in your template
  Allow SAM CLI IAM role creation [Y/n]: ENTER
  #Preserves the state of previously provisioned resources when an operation
fails
  Disable rollback [y/N]: ENTER
  HelloWorldFunction may not have authorization defined, Is this okay? [y/
N]: y
  Save arguments to configuration file [Y/n]: ENTER
  SAM configuration file [samconfig.toml]: ENTER
  SAM configuration environment [default]: ENTER

```

4. A continuación, AWS SAMCLI despliega su aplicación en. Nube de AWS Durante la implementación, el progreso se muestra en el símbolo del sistema. Las etapas principales de la implementación son las siguientes:
- En el caso de las aplicaciones con AWS Lambda Event Functions empaquetadas como un archivo.zip, el AWS SAMCLI paquete se comprime y se carga en un bucket de Amazon Simple Storage Service (Amazon S3). Si es necesario, la CLI de AWS SAM creará un nuevo bucket.
 - En el caso de las aplicaciones con las funciones de Lambda empaquetadas como imagen de contenedor, la CLI de AWS SAM carga la imagen en Amazon Elastic Container Registry (Amazon ECR). Si es necesario, la CLI de AWS SAM creará un repositorio nuevo.
 - AWS SAMCLICrea un conjunto de AWS CloudFormation cambios y despliega la aplicación como una pila. CloudFormation
 - AWS SAMCLIModifica la AWS SAM plantilla implementada con el nuevo CodeUri valor para las funciones de Lambda.

A continuación, se muestra un ejemplo de la salida de implementación de la CLI de AWS SAM:

```

Looking for resources needed for deployment:

Managed S3 bucket: aws-sam-cli-managed-default-samclisam-s3-demo-
bucket-1a4x26zbcdkqr
A different default S3 bucket can be set in samconfig.toml and auto
resolution of buckets turned off by setting resolve_s3=False

```

Parameter "stack_name=sam-app" in [default.deploy.parameters] is defined as a global parameter [default.global.parameters].

This parameter will be only saved under [default.global.parameters] in /Users/.../sam-app/samconfig.toml.

Saved arguments to config file

Running 'sam deploy' for future deployments will use the parameters saved above.

The above parameters can be changed by modifying samconfig.toml

Learn more about samconfig.toml syntax at

<https://docs.aws.amazon.com/serverless-application-model/latest/developerguide/serverless-sam-cli-config.html>

```

Uploading to sam-app-zip/da3c598813f1c2151579b73ad788cac8 262144 / 619839
(42.29%)Uploading to sam-app-zip/da3c598813f1c2151579b73ad788cac8 524288 / 619839
(84.58%)Uploading to sam-app-zip/da3c598813f1c2151579b73ad788cac8 619839 /
619839 (100.00%)

```

Deploying with following values

=====

```

Stack name           : sam-app
Region               : us-west-2
Confirm changeset    : True
Disable rollback     : False
Deployment s3 bucket : aws-sam-cli-managed-default-samclisam-s3-
demo-bucket-1a4x26zbcdkqr
Capabilities          : ["CAPABILITY_IAM"]
Parameter overrides  : {}
Signing Profiles      : {}

```

Initiating deployment

=====

```

Uploading to sam-app-zip/be84c20f868068e4dc4a2c11966edf2d.template 1212 /
1212 (100.00%)

```

Waiting for changeset to be created..

CloudFormation stack changeset

Operation	LogicalResourceId	ResourceType
Replacement		

+ Add	HelloWorldFunctionHell	AWS::Lambda::Permissio	N/A
	oWorldPermissionProd	n	
+ Add	HelloWorldFunctionRole	AWS::IAM::Role	N/A
+ Add	HelloWorldFunction	AWS::Lambda::Function	N/A
+ Add	ServerlessRestApiDeplo	AWS::ApiGateway::Deplo	N/A
	yment47fc2d5f9d	yment	
+ Add	ServerlessRestApiProdS	AWS::ApiGateway::Stage	N/A
	tage		
+ Add	ServerlessRestApi	AWS::ApiGateway::RestA	N/A
		pi	

Changeset created successfully. arn:aws:cloudformation:us-west-2:012345678910:changeSet/samcli-deploy1680559234/d9f58a77-98bc-41cd-b9f4-433a5a450d7a

Previewing CloudFormation changeset before deployment

=====

Deploy this changeset? [y/N]: **y**

2023-04-03 12:00:50 - Waiting for stack create/update to complete

CloudFormation events from stack operations (refresh every 5.0 seconds)

ResourceStatus	ResourceType	LogicalResourceId	
ResourceStatusReason			
CREATE_IN_PROGRESS	AWS::IAM::Role	HelloWorldFunctionRole	-
CREATE_IN_PROGRESS	AWS::IAM::Role	HelloWorldFunctionRole	Resource creation

Initiated				
CREATE_COMPLETE	AWS::IAM::Role	HelloWorldFunctionRole	-	
CREATE_IN_PROGRESS	AWS::Lambda::Function	HelloWorldFunction	-	
CREATE_IN_PROGRESS creation	AWS::Lambda::Function	HelloWorldFunction	Resource	
Initiated				
CREATE_COMPLETE	AWS::Lambda::Function	HelloWorldFunction	-	
CREATE_IN_PROGRESS	AWS::ApiGateway::RestA	ServerlessRestApi	-	
	pi			
CREATE_IN_PROGRESS creation	AWS::ApiGateway::RestA	ServerlessRestApi	Resource	
	pi			
Initiated				
CREATE_COMPLETE	AWS::ApiGateway::RestA	ServerlessRestApi	-	
	pi			
CREATE_IN_PROGRESS	AWS::Lambda::Permissio	HelloWorldFunctionHell	-	
	n	oWorldPermissionProd		
CREATE_IN_PROGRESS	AWS::ApiGateway::Deplo	ServerlessRestApiDeplo	-	
	yment	yment47fc2d5f9d		
CREATE_IN_PROGRESS creation	AWS::Lambda::Permissio	HelloWorldFunctionHell	Resource	
	n	oWorldPermissionProd		
Initiated				
CREATE_IN_PROGRESS creation	AWS::ApiGateway::Deplo	ServerlessRestApiDeplo	Resource	
	yment	yment47fc2d5f9d		
Initiated				
CREATE_COMPLETE	AWS::ApiGateway::Deplo	ServerlessRestApiDeplo	-	
	yment	yment47fc2d5f9d		

CREATE_IN_PROGRESS	AWS::ApiGateway::Stage	ServerlessRestApiProdStage	-
CREATE_IN_PROGRESS creation	AWS::ApiGateway::Stage	ServerlessRestApiProdStage	Resource
Initiated CREATE_COMPLETE	AWS::ApiGateway::Stage	ServerlessRestApiProdStage	-
CREATE_COMPLETE	AWS::Lambda::Permission	HelloWorldFunctionHelloWorldPermissionProd	-
CREATE_COMPLETE	AWS::CloudFormation::Stack	sam-app-stack	-

CloudFormation outputs from deployed stack

Outputs

Key	HelloWorldFunctionIamRole
Description	Implicit IAM Role created for Hello World function
Value	arn:aws:iam::012345678910:role/sam-app-zip-HelloWorldFunctionRole-11Z0GSCG28H0M
Key	HelloWorldApi
Description function	API Gateway endpoint URL for Prod stage for Hello World
Value hello/	https://njzfhdm1s0.execute-api.us-west-2.amazonaws.com/Prod/hello/


```
Key                HelloWorldFunction

Description        Hello World Lambda Function ARN

Value              arn:aws:lambda:us-west-2:012345678910:function:sam-app-
HelloWorldFunction-XPqNX4TBu7qn

-----

Successfully created/updated stack - sam-app-zip in us-west-2
```

5. Para ver la aplicación implementada, haga lo siguiente:

1. Abra la CloudFormation consola directamente con la URL. <https://console.aws.amazon.com/cloudformation>
2. Seleccione Stacks (Pilas).
3. Identifique su pila por el nombre de la aplicación y selecciónela.

Comprueba los cambios antes de la implementación

Puede configurarlo AWS SAMCLI para que muestre el conjunto de CloudFormation cambios y solicite una confirmación antes de implementarlo.

Para confirmar los cambios antes de la implementación

1. Durante `sam deploy --guided`, introduzca **Y** para confirmar los cambios antes de la implementación.

```
#Shows you resources changes to be deployed and require a 'Y' to initiate deploy
Confirm changes before deploy [Y/n]: Y
```

Como alternativa, puede modificar el archivo `samconfig.toml` con lo siguiente:

```
[default.deploy]
[default.deploy.parameters]
confirm_changeset = true
```

2. Durante la implementación, la AWS SAM de CLI le pedirá que confirme los cambios antes de la implementación. A continuación, se muestra un ejemplo:

```
Waiting for changeset to be created..
```

```
CloudFormation stack changeset
```

Operation	LogicalResourceId	ResourceType	
Replacement			
+ Add	HelloWorldFunctionHell	AWS::Lambda::Permissio	N/A
	oWorldPermissionProd	n	
+ Add	HelloWorldFunctionRole	AWS::IAM::Role	N/A
+ Add	HelloWorldFunction	AWS::Lambda::Function	N/A
+ Add	ServerlessRestApiDeplo	AWS::ApiGateway::Deplo	N/A
	yment47fc2d5f9d	yment	
+ Add	ServerlessRestApiProdS	AWS::ApiGateway::Stage	N/A
	tage		
+ Add	ServerlessRestApi	AWS::ApiGateway::RestA	N/A
		pi	

```
Changeset created successfully. arn:aws:cloudformation:us-
west-2:012345678910:changeSet/samcli-deploy1680559234/d9f58a77-98bc-41cd-
b9f4-433a5a450d7a
```

```
Previewing CloudFormation changeset before deployment
```

```
=====
```

```
Deploy this changeset? [y/N]: y
```

Especifica parámetros adicionales durante la implementación

Puedes especificar valores de parámetros adicionales para configurarlos durante la implementación. Para ello, modifique la plantilla de AWS SAM y configure el valor del parámetro durante la implementación.

Para especificar parámetros adicionales

1. Modifique la `Parameters` sección de la AWS SAM plantilla. A continuación, se muestra un ejemplo:

```
AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31
...
Globals:
...
Parameters:
  DomainName:
    Type: String
    Default: example
    Description: Domain name
```

2. Ejecute `sam deploy --guided`. El siguiente es un ejemplo de output:

```
sam-app $ sam deploy --guided

Configuring SAM deploy
=====

    Looking for config file [samconfig.toml] : Found
    Reading default arguments : Success

    Setting default arguments for 'sam deploy'
    =====
    Stack Name [sam-app-zip]: ENTER
    AWS Region [us-west-2]: ENTER
    Parameter DomainName [example]: ENTER
```

Configure la firma de código para sus funciones de eventos de Lambda

Puede configurar la firma de código para las funciones de Lambda en el momento de la implementación. Para ello, modifique la AWS SAM plantilla y configure la firma de código durante la implementación.

Para configurar la firma de código

1. Especifica `CodeSigningConfigArn` en tu AWS SAM plantilla. A continuación, se muestra un ejemplo:

```
AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31
...
Resources:
  HelloWorldFunction:
    Type: AWS::Serverless::Function
    Properties:
      CodeUri: hello_world/
      Handler: app.lambda_handler
      Runtime: python3.7
      CodeSigningConfigArn: arn:aws:lambda:us-east-1:111122223333:code-signing-
config:csc-12e12345db1234567
```

2. Ejecute `sam deploy --guided`. La CLI de AWS SAM le pedirá que configure la firma de código. El siguiente es un ejemplo de output:

```
#Found code signing configurations in your function definitions
Do you want to sign your code? [Y/n]: ENTER
#Please provide signing profile details for the following functions & layers
#Signing profile details for function 'HelloWorld'
Signing Profile Name:
Signing Profile Owner Account ID (optional):
#Signing profile details for layer 'MyLayer', which is used by functions
{'HelloWorld'}
Signing Profile Name:
Signing Profile Owner Account ID (optional):
```

Prácticas recomendadas

- Cuando se usa `sam deploy`, la CLI de AWS SAM implementa los artefactos de compilación de la aplicación ubicados en el directorio `.aws-sam`. Cuando realice cambios en los archivos originales de la aplicación, ejecútela con `sam build` para actualizar el directorio `.aws-sam` antes de implementarla.
- Al implementar una aplicación por primera vez, utilícela con `sam deploy --guided` para configurar los ajustes de implementación. Para las implementaciones posteriores, puede utilizar `sam deploy` para realizar la implementación con los parámetros configurados.

Opciones para una implementación de sam

A continuación, se indican las opciones más utilizadas para `sam deploy`. Para obtener una lista de todas las opciones, consulta [sam deploy](#).

Utiliza el flujo interactivo guiado para implementar la aplicación

Utiliza la opción `--guided` para configurar los ajustes de implementación de la aplicación mediante un flujo interactivo. A continuación, se muestra un ejemplo:

```
$ sam deploy --guided
```

La configuración de implementación de la aplicación se guarda en el archivo `samconfig.toml` del proyecto. Para obtener más información, consulta [Configurar los ajustes del proyecto](#).

Resolución de problemas

Para solucionar el problema AWS SAMCLI, consulte [AWS SAM CLI solución de problemas](#).

Ejemplos

Implemente una aplicación Hello World que contenga una función de eventos Lambda empaquetada como un archivo.zip

Para ver un ejemplo, consulta el tutorial [Paso 3: Implemente la aplicación en Nube de AWS](#) de la aplicación Hello World.

Implemente una aplicación Hello World que contenga una función de eventos de Lambda empaquetada como una imagen de contenedor

En primer lugar, cree la aplicación Hello World. `sam init` Durante el flujo interactivo, elija el tiempo de ejecución Python3.9 y el tipo de paquete Image.

```
$ sam init
...
Which template source would you like to use?
    1 - AWS Quick Start Templates
    2 - Custom Template Location
Choice: 1

Choose an AWS Quick Start application template
    1 - Hello World Example
    2 - Multi-step workflow
    ...
Template: 1

Use the most popular runtime and package type? (Python and zip) [y/N]: ENTER

Which runtime would you like to use?
    1 - aot.dotnet7 (provided.al2)
    ...
    15 - nodejs12.x
    16 - python3.9
    17 - python3.8
    ...
Runtime: 16

What package type would you like to use?
    1 - Zip
    2 - Image
Package type: 2

Based on your selections, the only dependency manager available is pip.
We will proceed copying the template using pip.
...
Project name [sam-app]: ENTER

-----
Generating application:
-----
```

```
Name: sam-app
Base Image: amazon/python3.9-base
Architectures: x86_64
Dependency Manager: pip
Output Directory: .
Configuration file: sam-app/samconfig.toml
```

Next steps can be found in the README file at sam-app/README.md

...

A continuación, vaya al directorio raíz `cd` del proyecto y ejecútelo con `sam build`. La CLI de AWS SAM crea la función de Lambda localmente usando Docker.

```
sam-app $ sam build
Building codeuri: /Users/.../sam-app runtime: None metadata: {'Dockerfile':
'Dockerfile', 'DockerContext': '/Users/.../sam-app/hello_world', 'DockerTag':
'python3.9-v1'} architecture: x86_64 functions: HelloWorldFunction
Building image for HelloWorldFunction function
Setting DockerBuildArgs: {} for HelloWorldFunction function
Step 1/5 : FROM public.ecr.aws/lambda/python:3.9
----> 0a5e3da309aa
Step 2/5 : COPY requirements.txt ./
----> abc4e82e85f9
Step 3/5 : RUN python3.9 -m pip install -r requirements.txt -t .
----> [Warning] The requested image's platform (linux/amd64) does not match the
detected host platform (linux/arm64/v8) and no specific platform was requested
----> Running in 43845e7aa22d
Collecting requests
  Downloading requests-2.28.2-py3-none-any.whl (62 kB)
##### 62.8/62.8 KB 829.5 kB/s eta 0:00:00
Collecting idna<4,>=2.5
  Downloading idna-3.4-py3-none-any.whl (61 kB)
##### 61.5/61.5 KB 2.4 MB/s eta 0:00:00
Collecting charset-normalizer<4,>=2
  Downloading charset_normalizer-3.1.0-cp39-cp39-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl (199 kB)
##### 199.2/199.2 KB 2.1 MB/s eta 0:00:00
Collecting certifi>=2017.4.17
  Downloading certifi-2022.12.7-py3-none-any.whl (155 kB)
##### 155.3/155.3 KB 10.2 MB/s eta 0:00:00
Collecting urllib3<1.27,>=1.21.1
  Downloading urllib3-1.26.15-py2.py3-none-any.whl (140 kB)
##### 140.9/140.9 KB 9.1 MB/s eta 0:00:00
```

```

Installing collected packages: urllib3, idna, charset-normalizer, certifi, requests
Successfully installed certifi-2022.12.7 charset-normalizer-3.1.0 idna-3.4
requests-2.28.2 urllib3-1.26.15
Removing intermediate container 43845e7aa22d
---> cab8ace899ce
Step 4/5 : COPY app.py ./
---> 4146f3cd69f2
Step 5/5 : CMD ["app.lambda_handler"]
---> [Warning] The requested image's platform (linux/amd64) does not match the
detected host platform (linux/arm64/v8) and no specific platform was requested
---> Running in f4131ddffb31
Removing intermediate container f4131ddffb31
---> d2f5180b2154
Successfully built d2f5180b2154
Successfully tagged helloworldfunction:python3.9-v1

```

Build Succeeded

```

Built Artifacts   : .aws-sam/build
Built Template    : .aws-sam/build/template.yaml

```

Commands you can use next

=====

```

[*] Validate SAM template: sam validate
[*] Invoke Function: sam local invoke
[*] Test Function in the Cloud: sam sync --stack-name {{stack-name}} --watch
[*] Deploy: sam deploy --guided

```

A continuación, ejecuta `sam deploy --guided` para crear e implementar la aplicación. La CLI de AWS SAM lo guía a través de la configuración de los ajustes de implementación. Luego, AWS SAMCLI despliega nuestra aplicación en. Nube de AWS

```
sam-app $ sam deploy --guided
```

Configuring SAM deploy

=====

```

    Looking for config file [samconfig.toml] : Found
    Reading default arguments : Success

```

```

    Setting default arguments for 'sam deploy'
    =====

```



```

Stack Name [sam-app]: ENTER
AWS Region [us-west-2]: ENTER
#Shows you resources changes to be deployed and require a 'Y' to initiate
deploy
Confirm changes before deploy [Y/n]: ENTER
#SAM needs permission to be able to create roles to connect to the resources in
your template
Allow SAM CLI IAM role creation [Y/n]: ENTER
#Preserves the state of previously provisioned resources when an operation
fails
Disable rollback [y/N]: ENTER
HelloWorldFunction may not have authorization defined, Is this okay? [y/N]: y
Save arguments to configuration file [Y/n]: ENTER
SAM configuration file [samconfig.toml]: ENTER
SAM configuration environment [default]: ENTER

Looking for resources needed for deployment:

Managed S3 bucket: aws-sam-cli-managed-default-samclisam-s3-demo-
bucket-1a4x26zbcdkqr
A different default S3 bucket can be set in samconfig.toml and auto resolution
of buckets turned off by setting resolve_s3=False

Parameter "stack_name=sam-app" in [default.deploy.parameters] is defined as a
global parameter [default.global.parameters].
This parameter will be only saved under [default.global.parameters] in /
Users/.../sam-app/samconfig.toml.

Saved arguments to config file
Running 'sam deploy' for future deployments will use the parameters saved
above.

The above parameters can be changed by modifying samconfig.toml
Learn more about samconfig.toml syntax at
https://docs.aws.amazon.com/serverless-application-model/latest/developerguide/
serverless-sam-cli-config.html

e95fc5e75742: Pushed
d8df51e7bdd7: Pushed
b1d0d7e0b34a: Pushed
0071317b94d8: Pushed
d98f98baf147: Pushed
2d244e0816c6: Pushed
eb2eeb1ebe42: Pushed
a5ca065a3279: Pushed

```

```
fe9e144829c9: Pushed
helloworldfunction-d2f5180b2154-python3.9-v1: digest:
sha256:cceb71401b47dc3007a7a1e1f2e0baf162999e0e6841d15954745ecc0c447533 size: 2206

    Deploying with following values
    =====
    Stack name                : sam-app
    Region                    : us-west-2
    Confirm changeset        : True
    Disable rollback         : False
    Deployment image repository :
                                {
                                    "HelloWorldFunction":
"012345678910.dkr.ecr.us-west-2.amazonaws.com/samapp7427b055/
helloworldfunction19d43fc4repo"
                                }
    Deployment s3 bucket      : aws-sam-cli-managed-default-samclisam-s3-demo-
bucket-1a4x26zbcdkqr
    Capabilities              : ["CAPABILITY_IAM"]
    Parameter overrides       : {}
    Signing Profiles          : {}

Initiating deployment
=====

HelloWorldFunction may not have authorization defined.
    Uploading to sam-app/682ad27c7cf7a17c7f77a1688b0844f2.template  1328 / 1328
(100.00%)

Waiting for changeset to be created..

CloudFormation stack changeset
-----
Operation                LogicalResourceId      ResourceType      Replacement
-----
+ Add                    HelloWorldFunctionHell  AWS::Lambda::Permissio  N/A
                           oWorldPermissionProd    n
+ Add                    HelloWorldFunctionRole  AWS::IAM::Role      N/A
```

+ Add	HelloWorldFunction	AWS::Lambda::Function	N/A
+ Add	ServerlessRestApiDeployment47fc2d5f9d	AWS::ApiGateway::Deployment	N/A
+ Add	ServerlessRestApiProdStage	AWS::ApiGateway::Stage	N/A
+ Add	ServerlessRestApi	AWS::ApiGateway::RestApi	N/A

Changeset created successfully. arn:aws:cloudformation:us-west-2:012345678910:changeSet/samcli-deploy1680634124/0fffd4faf-2e2b-487e-b9e0-9116e8299ac4			
Previewing CloudFormation changeset before deployment			
=====			
Deploy this changeset? [y/N]: y			
2023-04-04 08:49:15 - Waiting for stack create/update to complete			
CloudFormation events from stack operations (refresh every 5.0 seconds)			

ResourceStatus	ResourceType	LogicalResourceId	
ResourceStatusReason			

CREATE_IN_PROGRESS Initiated	AWS::CloudFormation::Stack	sam-app	User
CREATE_IN_PROGRESS	AWS::IAM::Role	HelloWorldFunctionRole	-
CREATE_IN_PROGRESS creation	AWS::IAM::Role	HelloWorldFunctionRole	Resource Initiated

CREATE_COMPLETE	AWS::IAM::Role	HelloWorldFunctionRole	-
CREATE_IN_PROGRESS	AWS::Lambda::Function	HelloWorldFunction	-
CREATE_IN_PROGRESS creation	AWS::Lambda::Function	HelloWorldFunction	Resource Initiated
CREATE_COMPLETE	AWS::Lambda::Function	HelloWorldFunction	-
CREATE_IN_PROGRESS	AWS::ApiGateway::RestA pi	ServerlessRestApi	-
CREATE_IN_PROGRESS creation	AWS::ApiGateway::RestA pi	ServerlessRestApi	Resource Initiated
CREATE_COMPLETE	AWS::ApiGateway::RestA pi	ServerlessRestApi	-
CREATE_IN_PROGRESS	AWS::Lambda::Permissio n	HelloWorldFunctionHell oWorldPermissionProd	-
CREATE_IN_PROGRESS	AWS::ApiGateway::Deplo yment	ServerlessRestApiDeplo yment47fc2d5f9d	-
CREATE_IN_PROGRESS creation	AWS::Lambda::Permissio n	HelloWorldFunctionHell oWorldPermissionProd	Resource Initiated
CREATE_IN_PROGRESS creation	AWS::ApiGateway::Deplo yment	ServerlessRestApiDeplo yment47fc2d5f9d	Resource Initiated
CREATE_COMPLETE	AWS::ApiGateway::Deplo yment	ServerlessRestApiDeplo yment47fc2d5f9d	-
CREATE_IN_PROGRESS	AWS::ApiGateway::Stage	ServerlessRestApiProdS	-

		stage	
CREATE_IN_PROGRESS creation	AWS::ApiGateway::Stage	ServerlessRestApiProdS	Resource
		stage	Initiated
CREATE_COMPLETE	AWS::ApiGateway::Stage	ServerlessRestApiProdS	-
		stage	
CREATE_COMPLETE	AWS::Lambda::Permissio	HelloWorldFunctionHell	-
	n	oWorldPermissionProd	
CREATE_COMPLETE	AWS::CloudFormation::S	sam-app	-
	tack		

CloudFormation outputs from deployed stack

Outputs

Key	HelloWorldFunctionIamRole
Description	Implicit IAM Role created for Hello World function
Value	arn:aws:iam::012345678910:role/sam-app-HelloWorldFunctionRole-JFML1J0KHJ71
Key	HelloWorldApi
Description	API Gateway endpoint URL for Prod stage for Hello World function
Value	https://endlwiqqod.execute-api.us-west-2.amazonaws.com/Prod/hello/
Key	HelloWorldFunction
Description	Hello World Lambda Function ARN

```
Value                arn:aws:lambda:us-west-2:012345678910:function:sam-app-  
HelloWorldFunction-  
kyg6Y2iNRUPg
```

Successfully created/updated stack - sam-app in us-west-2

Más información

Para obtener más información sobre el uso del AWS SAMCLI `sam deploy` comando, consulte lo siguiente:

- [El AWS SAM taller completo: Módulo 3: Implementación manual](#): aprenda a crear, empaquetar e implementar una aplicación sin servidor mediante el AWS SAMCLI.

Opciones para implementar su aplicación con AWS SAM

Con él AWS SAM, puede implementar su aplicación manualmente y automatizar las implementaciones. Use la CLI de AWS SAM para implementar la aplicación manualmente. Para automatizar la implementación, use canalizaciones y un sistema de implementación e integración continuas (CI/CD). En los temas de esta sección se proporciona información sobre ambos enfoques.

Temas

- [Cómo usar la CLI de AWS SAM para realizar una implementación manual](#)
- [Implemente con CI/CD sistemas y canalizaciones](#)
- [Implementación gradual](#)
- [Solución de problemas de las implementaciones mediante la CLI de AWS SAM](#)
- [Más información](#)

Cómo usar la CLI de AWS SAM para realizar una implementación manual

Después de desarrollar y testar la aplicación sin servidor a nivel local, puedes implementarla mediante el comando [sam deploy](#).

Para AWS SAM guiarlo a lo largo de la implementación con instrucciones, especifique el `--guided` indicador. Al especificar este indicador, el comando `sam deploy` comprime los artefactos

de la aplicación y los carga en Amazon Simple Storage Service (Amazon S3) (para archivos de archivos .zip) o en Amazon Elastic Container Registry (Amazon ECR) (para imágenes de contenedores). A continuación, el comando despliega la aplicación en la AWS nube.

Ejemplo:

```
# Deploy an application using prompts:
sam deploy --guided
```

Implemente con CI/CD sistemas y canalizaciones

AWS SAM le ayuda a automatizar la implementación mediante canalizaciones y un sistema de integración e implementación continuas (CI/CD). AWS SAM se puede utilizar para crear canalizaciones y simplificar las CI/CD tareas de las aplicaciones sin servidor. Varios CI/CD sistemas admiten la AWS SAM creación de imágenes de contenedores y AWS SAM también proporcionan un conjunto de plantillas de canalización predeterminadas para varios CI/CD sistemas que encapsulan las mejores prácticas AWS de implementación.

Para obtener más información, consulte [Uso CI/CD de sistemas y canalizaciones para implementar con AWS SAM](#).

Implementación gradual

Si desea implementar la AWS SAM aplicación de forma gradual en lugar de hacerlo de una sola vez, puede especificar las configuraciones de implementación que se AWS CodeDeploy proporcionan. Para obtener más información, consulte [Trabajar con configuraciones de despliegue CodeDeploy en la Guía del AWS CodeDeploy usuario](#).

Para obtener información sobre cómo configurar AWS SAM la aplicación para que se despliegue de forma gradual, consulte [Implementación gradual de aplicaciones sin servidor con AWS SAM](#).

Solución de problemas de las implementaciones mediante la CLI de AWS SAM

Error de la CLI de AWS SAM: «no se cumplen las restricciones de seguridad»

Al ejecutar `sam deploy --guided`, se le preguntará `HelloWorldFunction may not have authorization defined, Is this okay? [y/N]`. Si respondes a esta solicitud con **N** (la respuesta predeterminada), verás el siguiente error:

```
Error: Security Constraints Not Satisfied
```

El mensaje le informa de que la aplicación que va a implementar podría tener una API de Amazon API Gateway de acceso público configurada sin autorización. Al responder **N** a esta pregunta, está diciendo que esto no está bien.

Para resolver este problema, dispones de las siguientes opciones:

- Configura tu aplicación con autorización. Para obtener más información acerca de la autorización, consulta [Controle el acceso a la API con su AWS SAM plantilla](#).
- Responde a esta pregunta **Y** para indicar que estás de acuerdo con la implementación de una aplicación que tenga una API de API Gateway configurada sin autorización.

Más información

Para ver ejemplos prácticos de la implementación de aplicaciones sin servidor, consulte lo siguiente de The Complete AWS SAM Workshop:

- [Módulo 3: Implementación manual](#): aprenda a crear, empaquetar e implementar una aplicación sin servidor mediante AWS SAMCLI
- [Módulo 4: CI/CD](#): aprenda a automatizar las fases de creación, empaquetado e implementación mediante la creación de una canalización de integración y entrega continuas (CI/CD).

Uso CI/CD de sistemas y canalizaciones para implementar con AWS SAM

AWS SAM ayuda a las organizaciones a crear canalizaciones para sus CI/CD sistemas preferidos, de modo que puedan aprovechar los beneficios de ellos CI/CD con un mínimo esfuerzo, como acelerar la frecuencia de implementación, acortar el tiempo de espera de los cambios y reducir los errores de implementación.

AWS SAM simplifica CI/CD las tareas de las aplicaciones sin servidor con la ayuda de la creación de imágenes de contenedores. Las imágenes que se AWS SAM proporcionan incluyen las herramientas de compilación AWS SAMCLI y las herramientas para varios tiempos de ejecución compatibles AWS Lambda . Esto facilita la creación y el empaquetado de aplicaciones sin servidor mediante el AWS

SAMCLI Estas imágenes también reducen la necesidad de que los equipos creen y administren sus propias imágenes para CI/CD los sistemas. Para obtener más información sobre la AWS SAM creación de imágenes de contenedores, consulte [Repositorios de imágenes para AWS SAM](#).

Varios CI/CD sistemas admiten la AWS SAM creación de imágenes de contenedores. CI/CD El sistema que debe utilizar depende de varios factores. Estos incluyen si la aplicación utiliza uno o varios tiempos de ejecución, o si desea compilar la aplicación dentro de una imagen del contenedor o directamente en una máquina host, ya sea una máquina virtual (VM) o un host bare metal.

AWS SAM también proporciona un conjunto de plantillas de canalización predeterminadas para varios CI/CD sistemas que resumen las mejores prácticas AWS de implementación. Estas plantillas de canalización predeterminadas utilizan formatos de configuración de JSON/YAML canalización estándar, y las prácticas recomendadas integradas ayudan a realizar despliegues con múltiples cuentas y múltiples regiones, y a verificar que las canalizaciones no puedan realizar cambios involuntarios en la infraestructura.

Tiene dos opciones principales AWS SAM para implementar sus aplicaciones sin servidor: 1) Modificar la configuración de canalización existente para usar AWS SAMCLI comandos, o 2) generar un ejemplo de configuración de CI/CD canalización que pueda usar como punto de partida para su propia aplicación.

Temas

- [¿Qué es una canalización?](#)
- [Cómo AWS SAM carga archivos locales en el momento de la implementación](#)
- [Genera una CI/CD canalización inicial con AWS SAM](#)
- [Cómo personalizar las canalizaciones iniciales con AWS SAM](#)
- [Automatice la implementación de su AWS SAM aplicación](#)
- [Cómo utilizar la autenticación OIDC con canalizaciones AWS SAM](#)

¿Qué es una canalización?

Una canalización es una secuencia automática de pasos que se llevan a cabo para lanzar una nueva versión de una aplicación. [Con AWS SAMél, puede usar muchos CI/CD sistemas comunes para implementar sus aplicaciones, incluidos Jenkins AWS CodePipeline, GitLab CI/CD y Actions. GitHub](#)

Las plantillas de Pipeline incluyen las mejores prácticas de AWS implementación para facilitar las implementaciones con múltiples cuentas y múltiples regiones. AWS Los entornos como los de

desarrollo y producción suelen existir en cuentas diferentes. AWS Esto permite a los equipos de desarrollo configurar canalizaciones de implementación seguras, sin realizar cambios involuntarios en la infraestructura.

También puede proporcionar sus propias plantillas de canalización personalizadas para ayudar a estandarizar las canalizaciones en todos los equipos de desarrollo.

Cómo AWS SAM carga archivos locales en el momento de la implementación

Cuando despliega su aplicación en Nube de AWS, AWS CloudFormation requiere que sus archivos locales se carguen primero en un AWS servicio accesible, como Amazon Simple Storage Service (Amazon S3). Esto incluye los archivos locales a los que hace referencia tu AWS SAM plantilla. Para cumplir con este requisito, la CLI de AWS SAM hace lo siguiente cuando utiliza el comando `sam deploy` o `sam package`:

1. Carga automáticamente sus archivos locales a un AWS servicio accesible.
2. Actualiza automáticamente la plantilla de la aplicación para hacer referencia a la nueva ruta del archivo.

Temas

- [Demostración: utilice el AWS SAM CLI para cargar el código de la función de evento Lambda](#)
- [Casos de uso admitidos](#)
- [Más información](#)

Demostración: utilice el AWS SAM CLI para cargar el código de la función de evento Lambda

En esta demostración, inicializamos la aplicación Hello World de ejemplo con un tipo de `paquete.zip` para nuestra función de eventos Lambda. Utilizamos la CLI de AWS SAM para cargar automáticamente el código de la función de Lambda en Amazon S3 y hacer referencia a su nueva ruta en la plantilla de aplicación.

Primero, ejecutamos `sam init` para inicializar la aplicación Hello World.

```
$ sam init
...
```

```

Which template source would you like to use?
    1 - AWS Quick Start Templates
    2 - Custom Template Location
Choice: 1

Choose an AWS Quick Start application template
    1 - Hello World Example
    2 - Multi-step workflow
    ...
Template: 1

Use the most popular runtime and package type? (Python and zip) [y/N]: y

Would you like to enable X-Ray tracing on the function(s) in your application? [y/N]: ENTER

Would you like to enable monitoring using CloudWatch Application Insights?
For more info, please view https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/cloudwatch-application-insights.html [y/N]: ENTER

Project name [sam-app]: demo

-----
Generating application:
-----
Name: demo
Runtime: python3.9
Architectures: x86_64
Dependency Manager: pip
Application Template: hello-world
Output Directory: .
Configuration file: demo/samconfig.toml

...

```

El código de la función de Lambda está organizado en el subdirectorio `hello_world` de nuestro proyecto.

```

demo
### README.md
### hello_world
#   ### __init__.py
#   ### app.py

```

```
#   ### requirements.txt
### template.yaml
### tests
```

En nuestra AWS SAM plantilla, hacemos referencia a la ruta local al código de nuestra función Lambda mediante la `CodeUri` propiedad.

```
AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31
...
Resources:
  HelloWorldFunction:
    Type: AWS::Serverless::Function # More info about Function Resource:
    https://github.com/awslabs/serverless-application-model/blob/master/
versions/2016-10-31.md#awsserverlessfunction
    Properties:
      CodeUri: hello_world/
      Handler: app.lambda_handler
      Runtime: python3.9
      ...
```

A continuación, ejecuta `sam build` para crear e implementar la aplicación.

```
$ sam build
Starting Build use cache
Manifest file is changed (new hash: 3298f13049d19cffaa37ca931dd4d421) or dependency
folder (.aws-sam/deps/7896875f-9bcc-4350-8adb-2c1d543627a1) is missing for
(HelloWorldFunction), downloading dependencies and copying/building source
Building codeuri: /Users/.../demo/hello_world runtime: python3.9 metadata: {}
architecture: x86_64 functions: HelloWorldFunction
Running PythonPipBuilder:Cleanup
Running PythonPipBuilder:ResolveDependencies
Running PythonPipBuilder:CopySource
Running PythonPipBuilder:CopySource

Build Succeeded

Built Artifacts   : .aws-sam/build
Built Template    : .aws-sam/build/template.yaml
...
```

A continuación, ejecuta `sam deploy --guided` para crear e implementar la aplicación.

```
$ sam deploy --guided
```

```
Configuring SAM deploy
```

```
=====
```

```
Looking for config file [samconfig.toml] : Found
```

```
Reading default arguments : Success
```

```
Setting default arguments for 'sam deploy'
```

```
=====
```

```
Stack Name [demo]: ENTER
```

```
AWS Region [us-west-2]: ENTER
```

```
#Shows you resources changes to be deployed and require a 'Y' to initiate
deploy
```

```
Confirm changes before deploy [Y/n]: n
```

```
#SAM needs permission to be able to create roles to connect to the resources in
your template
```

```
Allow SAM CLI IAM role creation [Y/n]: ENTER
```

```
#Preserves the state of previously provisioned resources when an operation
fails
```

```
Disable rollback [y/N]: ENTER
```

```
HelloWorldFunction may not have authorization defined, Is this okay? [y/N]: y
```

```
Save arguments to configuration file [Y/n]: ENTER
```

```
SAM configuration file [samconfig.toml]: ENTER
```

```
SAM configuration environment [default]: ENTER
```

```
Looking for resources needed for deployment:
```

```
...
```

```
Saved arguments to config file
```

```
Running 'sam deploy' for future deployments will use the parameters saved
above.
```

```
The above parameters can be changed by modifying samconfig.toml
```

```
Learn more about samconfig.toml syntax at
```

```
https://docs.aws.amazon.com/serverless-application-model/latest/developerguide/
serverless-sam-cli-config.html
```

```
File with same data already exists at demo/da3c598813f1c2151579b73ad788cac8, skipping
upload
```

```
Deploying with following values
```

```
=====
```

```
Stack name : demo
```

```
Region : us-west-2
```

```

    Confirm changeset          : False
    Disable rollback          : False
    Deployment s3 bucket      : aws-sam-cli-managed-default-samclisam-s3-demo-
bucket-1a4x26zbcdkqr
    Capabilities              : ["CAPABILITY_IAM"]
    Parameter overrides       : {}
    Signing Profiles          : {}

```

Initiating deployment

=====

...

Waiting for changeset to be created..

CloudFormation stack changeset

Operation	LogicalResourceId	ResourceType	Replacement
+ Add	HelloWorldFunctionHell	AWS::Lambda::Permissio	N/A
	oWorldPermissionProd	n	
+ Add	HelloWorldFunctionRole	AWS::IAM::Role	N/A
...			

Changeset created successfully. arn:aws:cloudformation:us-west-2:012345678910:changeSet/samcli-deploy1680906292/1164338d-72e7-4593-a372-f2b3e67f542f

2023-04-07 12:24:58 - Waiting for stack create/update to complete

CloudFormation events from stack operations (refresh every 5.0 seconds)

ResourceStatus	ResourceType	LogicalResourceId	ResourceStatusReason
CREATE_IN_PROGRESS	AWS::IAM::Role	HelloWorldFunctionRole	-
CREATE_IN_PROGRESS	AWS::IAM::Role	HelloWorldFunctionRole	Resource creation
			Initiated
...			

```
-----
CloudFormation outputs from deployed stack
-----
```

```
Outputs
```

```
-----
Key                HelloWorldFunctionIamRole

Description        Implicit IAM Role created for Hello World function

Value              arn:aws:iam::012345678910:role/demo-HelloWorldFunctionRole-
VQ4CU7UY7S2K

Key                HelloWorldApi

Description        API Gateway endpoint URL for Prod stage for Hello World function

Value              https://satnon55e9.execute-api.us-west-2.amazonaws.com/Prod/hello/

Key                HelloWorldFunction

Description        Hello World Lambda Function ARN

Value              arn:aws:lambda:us-west-2:012345678910:function:demo-
HelloWorldFunction-G14inKTmSQvK
-----
```

```
Successfully created/updated stack - demo in us-west-2
```

Durante la implementación, la CLI de AWS SAM carga automáticamente el código de la función de Lambda a Amazon S3 y actualiza la plantilla. Nuestra plantilla modificada en la CloudFormation consola refleja la ruta del bucket de Amazon S3.

```
AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31
...
Resources:
  HelloWorldFunction:
    Type: AWS::Serverless::Function
    Properties:
```

```

    CodeUri: s3://aws-sam-cli-managed-default-samclisam-s3-demo-bucket-1a4x26zbcdkqr/
demo/da3c598813f1c2151579b73ad788cac8
    Handler: app.lambda_handler
    ...

```

Casos de uso admitidos

AWS SAMCLI Puede facilitar automáticamente este proceso para varios tipos de archivos, tipos de CloudFormation recursos y CloudFormation macros.

Tipos de archivos

Se admiten archivos de aplicaciones e imágenes de Docker.

CloudFormation tipos de recursos

La siguiente es una lista de los tipos de recursos compatibles y sus propiedades:

Recurso	Propiedades
AWS::ApiGateway::RestApi	BodyS3Location
AWS::ApiGatewayV2::Api	BodyS3Location
AWS::AppSync::FunctionConfiguration	CodeS3Location RequestMappingTemplateS3Location ResponseMappingTemplateS3Location
AWS::AppSync::GraphQLSchema	DefinitionS3Location
AWS::AppSync::Resolver	CodeS3Location RequestMappingTemplateS3Location ResponseMappingTemplateS3Location

Recurso	Propiedades
AWS::CloudFormation::ModuleVersion	ModulePackage
AWS::CloudFormation::ResourceVersion	SchemaHandlerPackage
AWS::ECR::Repository	RepositoryName
AWS::ElasticBeanstalk::ApplicationVersion	SourceBundle
AWS::Glue::Job	Command.ScriptLocation
AWS::Lambda::Function	Code Code.ImageUri
AWS::Lambda::LayerVersion	Content
AWS::Serverless::Api	DefinitionUri
AWS::Serverless::Function	CodeUri ImageUri
AWS::Serverless::GraphQLApi	SchemaUri Function.CodeUri Resolver.CodeUri
AWS::Serverless::HttpApi	DefinitionUri
AWS::Serverless::LayerVersion	ContentUri
AWS::Serverless::StateMachine	DefinitionUri

Recurso	Propiedades
<code>AWS::StepFunctions::StateMachine</code>	<code>DefinitionS3Location</code>

CloudFormation macros

Se admiten los archivos a los que se hace referencia mediante la macro de transformación de `AWS::Include`.

Más información

Para obtener más información sobre la `AWS::Include` transformación, consulte la [AWS::Include transformación](#) en la Guía del AWS CloudFormation usuario.

Para ver un ejemplo del uso de la `AWS::Include` transformación en una AWS SAM plantilla, consulta el patrón [API HTTP a SQS de API Gateway](#) en Serverless Land.

Genera una CI/CD canalización inicial con AWS SAM

Cuando esté listo para automatizar la implementación, puede usar una de las plantillas AWS SAM de canalización iniciales para generar una canalización de implementación para el CI/CD sistema que elija usar. La canalización de implementación es lo que configura y usa para automatizar la implementación de su aplicación sin servidor. Hay una plantilla de canalización de inicio preconfigurada para ayudarle a configurar rápidamente la canalización de implementación de la aplicación sin servidor.

Con una plantilla de canalización de inicio, puede generar canalizaciones en cuestión de minutos con el comando [sam pipeline init](#).

Las plantillas de canalización iniciales utilizan la JSON/YAML sintaxis habitual del CI/CD sistema e incorporan las mejores prácticas, como la gestión de los artefactos en varias cuentas y regiones y el uso de la cantidad mínima de permisos necesarios para implementar la aplicación. [Actualmente, la AWS SAM CLI admite la generación de configuraciones de CI/CD canalización iniciales para AWS CodePipeline las canalizaciones de Jenkins, GitLab CI/CD, GitHub Actions y Bitbucket.](#)

Estas son las tareas de alto nivel que debes realizar para generar una configuración de canalización inicial:

1. Cree recursos de infraestructura: su canalización requiere ciertos AWS recursos, por ejemplo, el usuario y las funciones de IAM con los permisos necesarios, un bucket de Amazon S3 y, opcionalmente, un repositorio de Amazon ECR.
2. Conecta tu repositorio de Git con tu CI/CD sistema: tu CI/CD sistema necesita saber qué repositorio de Git activará la canalización para que se ejecute. Ten en cuenta que este paso puede no ser necesario, dependiendo de la combinación de repositorio y CI/CD sistema de Git que utilices.
3. Genera tu configuración de canalización: este paso genera una configuración de canalización inicial que incluye dos etapas de implementación.
4. Confirma la configuración de la canalización en tu repositorio de Git: este paso es necesario para garantizar que el CI/CD sistema conozca la configuración de la canalización y se ejecute cuando se confirmen los cambios.

Una vez que hayas generado la configuración de canalización inicial y la hayas asignado a tu repositorio de Git, cada vez que alguien realice un cambio de código en ese repositorio, tu canalización se activará para que se ejecuta automáticamente.

El orden de estos pasos y los detalles de cada uno de ellos varían en función del sistema de CI/CD:

- Si está utilizando AWS CodePipeline, consulte [Generando una tubería de arranque para AWS CodePipeline en AWS SAM](#).
- Si utiliza Jenkins, GitLab CI/CD, GitHub Actions o Bitbucket Pipelines, consulte. [Se usa AWS SAM para generar canalizaciones iniciales para Jenkins, GitLab CI/CD, Actions y Bitbucket Pipelines GitHub](#)

Generando una tubería de arranque para AWS CodePipeline en AWS SAM

Para generar una configuración de canalización inicial para AWS CodePipeline, realice las siguientes tareas en este orden:

1. Crea recursos de infraestructura
2. Genere la configuración de la canalización
3. Confirme la configuración de su canalización en Git
4. Conecta tu repositorio de Git con tu CI/CD sistema

 Note

El siguiente procedimiento utilizas dos comandos CLI de AWS SAM, [sam pipeline bootstrap](#) y [sam pipeline init](#). La razón por la que hay dos comandos es para gestionar el caso de uso en el que los administradores (es decir, los usuarios que necesitan permiso para configurar los AWS recursos de infraestructura, como los usuarios y las funciones de IAM) tienen más permisos que los desarrolladores (es decir, los usuarios que solo necesitan permiso para configurar canalizaciones individuales, pero no los AWS recursos de infraestructura necesarios).

Paso 1: cree recursos de infraestructura

Las canalizaciones que se utilizan AWS SAM requieren determinados AWS recursos, como un usuario de IAM y las funciones con los permisos necesarios, un bucket de Amazon S3 y, opcionalmente, un repositorio de Amazon ECR. Debe tener un conjunto de recursos de infraestructura para cada etapa de implementación de la canalización.

Puedes ejecutar el siguiente comando para editar o agregar esta configuración:

```
sam pipeline bootstrap
```

 Note

Ejecuta el comando anterior para cada etapa de implementación de su canalización.

Paso 2: genere la configuración de la canalización

Para generar la configuración de la canalización, ejecuta el siguiente comando:

```
sam pipeline init
```

Paso 3: confirme la configuración de su canalización en el repositorio de Git

Este paso es necesario para garantizar que el CI/CD sistema conozca la configuración de la canalización y se ejecute cuando se realicen los cambios.

Paso 4: Conecta tu repositorio de Git con tu CI/CD sistema

Pues ahora AWS CodePipeline puedes crear la conexión ejecutando el siguiente comando:

```
sam deploy -t codepipeline.yaml --stack-name <pipeline-stack-name> --  
capabilities=CAPABILITY_IAM --region <region-X>
```

Si utilizas GitHub Bitbucket, después de ejecutar el sam deploy comando anteriormente, completa la conexión siguiendo los pasos descritos en Para completar una conexión, que se encuentran en el tema Actualizar una conexión [pendiente](#) de la guía del usuario de la consola de Developer Tools. Además, guarda una copia del resultado CodeStarConnectionArn del sam deploy comando, ya que la necesitarás si quieres usarla AWS CodePipeline con otra rama que no sea main.

Configuración de otras ramificaciones

De forma predeterminada, AWS CodePipeline usa la main rama con AWS SAM. Si quieres utilizar una ramificación que no sea main, debes volver a ejecutar el comando sam deploy. Ten en cuenta que, en función del repositorio de Git que utilices, es posible que también tengas que proporcionar el CodeStarConnectionArn:

```
# For GitHub and Bitbucket  
sam deploy -t codepipeline.yaml --stack-name <feature-pipeline-stack-name> --  
capabilities=CAPABILITY_IAM --parameter-overrides="FeatureGitBranch=<branch-name>  
CodeStarConnectionArn=<codestar-connection-arn>"  
  
# For AWS CodeCommit  
sam deploy -t codepipeline.yaml --stack-name <feature-pipeline-stack-name> --  
capabilities=CAPABILITY_IAM --parameter-overrides="FeatureGitBranch=<branch-name>"
```

Más información

Para ver un ejemplo práctico de cómo configurar una CI/CD canalización, consulta [CI/CD AWS CodePipeline en The Complete](#) Workshop. AWS SAM

Se usa AWS SAM para generar canalizaciones iniciales para Jenkins, GitLab CI/CD, Actions y Bitbucket Pipelines GitHub

Para generar una configuración de canalización inicial para Jenkins, GitLab CI/CD, GitHub Actions o Bitbucket Pipelines, realiza las siguientes tareas en este orden:

1. Crea recursos de infraestructura
2. Conecta tu repositorio de Git con tu CI/CD sistema
3. Crea objetos de credenciales
4. Genera la configuración de la canalización
5. Confirma su configuración de la canalización en el repositorio de Git

Note

El siguiente procedimiento utilizas dos comandos CLI de AWS SAM, [sam pipeline bootstrap](#) y [sam pipeline init](#). La razón por la que hay dos comandos es para gestionar el caso de uso en el que los administradores (es decir, los usuarios que necesitan permiso para configurar los AWS recursos de infraestructura, como los usuarios y las funciones de IAM) tienen más permisos que los desarrolladores (es decir, los usuarios que solo necesitan permiso para configurar canalizaciones individuales, pero no los AWS recursos de infraestructura necesarios).

Paso 1: cree recursos de infraestructura

Las canalizaciones que se utilizan AWS SAM requieren determinados AWS recursos, como un usuario de IAM y las funciones con los permisos necesarios, un bucket de Amazon S3 y, opcionalmente, un repositorio de Amazon ECR. Debe tener un conjunto de recursos de infraestructura para cada etapa de implementación de la canalización.

Puedes ejecutar el siguiente comando para editar o agregar esta configuración:

```
sam pipeline bootstrap
```

Note

Ejecuta el comando anterior para cada etapa de implementación de su canalización.

Debe capturar AWS las credenciales (el identificador de clave y la clave secreta) de los usuarios de la canalización para cada etapa de despliegue de la canalización, ya que son necesarias para los pasos siguientes.

Paso 2: Conecta tu repositorio de Git con tu CI/CD sistema

Es necesario conectar el repositorio de Git al CI/CD sistema para que el CI/CD sistema pueda acceder al código fuente de la aplicación para compilaciones e implementaciones.

Note

Puedes omitir este paso si utilizas una de las siguientes combinaciones, ya que la conexión se realiza automáticamente:

1. GitHub Acciones con el repositorio GitHub
2. GitLab CI/CD con repositorio GitLab
3. Bitbucket Pipelines con repositorio de Bitbucket

Para conectar el repositorio de Git con el CI/CD sistema, realiza una de las siguientes acciones:

- Si utilizas Jenkins, consulta la [documentación de Jenkins](#) para «añadir un origen de ramificación».
- Si utilizas GitLab CI/CD y otro repositorio de Git GitLab, consulta la [GitLabdocumentación](#) para «conectar un repositorio externo».

Paso 3: crear objetos de credenciales

Cada CI/CD sistema tiene su propia forma de administrar las credenciales necesarias para que el CI/CD sistema acceda a tu repositorio de Git.

Para crear los objetos de credenciales necesarios, haz una de las siguientes acciones:

- Si utilizas Jenkins, cree una «credencial» única que almacene tanto el identificador como la contraseña. Sigue las instrucciones del blog [Cómo crear una canalización con Jenkins con AWS SAM](#), en la sección Configurar Jenkins. Necesitarás el «identificador de credencial» para el siguiente paso.
- Si utilizas GitLab CI/CD, crea dos «variables protegidas», una para cada identificador de clave y clave secreta. Siga las instrucciones de la [GitLab documentación](#): necesitará dos «claves variables» para el siguiente paso.
- Si utilizas GitHub Actions, crea dos «secretos cifrados», uno para cada clave y una clave secreta. Sigue las instrucciones de la [GitHubdocumentación](#): necesitarás dos «nombres secretos» para el siguiente paso.

- Si utilizas Bitbucket Pipelines, crea dos «variables seguras», una para cada identificador y contraseña. Sigue las instrucciones de las [variables y secretos](#); necesitarás dos «nombres secretos» para el siguiente paso.

Paso 4: genere la configuración de la canalización

Para generar la configuración de la canalización, ejecuta el siguiente comando. Necesitará introducir el objeto de credenciales que creaste en el paso anterior:

```
sam pipeline init
```

Paso 5: confirma la configuración de su canalización en el repositorio de Git

Este paso es necesario para garantizar que el CI/CD sistema conozca la configuración de la canalización y se ejecute cuando se realicen los cambios.

Más información

Para ver un ejemplo práctico de cómo configurar una CI/CD canalización mediante el uso de este GitHub Actions método, consulte [CI/CD GitHub en The Complete](#) Workshop. AWS SAM

Cómo personalizar las canalizaciones iniciales con AWS SAM

Como CI/CD administrador, es posible que desee personalizar una plantilla de canalización inicial y las instrucciones guiadas asociadas, para que los desarrolladores de su organización puedan usar para crear configuraciones de canalización.

La CLI de AWS SAM utiliza plantillas de Cookiecutter para crear plantillas de inicio. Para obtener más información sobre las plantillas de cookie cutter, [Cookiecutter](#).

También puedes personalizar las instrucciones que la CLI de AWS SAM muestra a los usuarios al crear configuraciones de canalización mediante el comando `sam pipeline init`. Para personalizar las solicitudes de usuario, haga lo siguiente:

1. Crea un archivo de **questions.json**: el archivo `questions.json` debes estar en la raíz del repositorio del proyecto. Este es el mismo directorio que el del archivo `cookiecutter.json`. Para ver el esquema del archivo `questions.json`, consulta [questions.json.schema](#). Para ver un ejemplo de archivo `questions.json`, consulta [questions.json](#).
2. Asigna las claves de preguntas a los nombres de los cookiecutter: cada objeto del archivo `questions.json` necesita una claves que coincida con el nombre de la plantilla del cookiecutter.

Esta coincidencia de claves es la forma en que la CLI de AWS SAM asigna las respuestas de las preguntas del usuario a la plantilla cookie cutter. Para obtener ejemplos de esta coincidencia de claves, consulta la sección [Archivos de ejemplo](#) más adelante en este tema.

3. Crea un archivo **metadata.json**: declara el número de etapas que tendrá la canalización en el archivo de metadata.json. El número de etapas indica al comando `sam pipeline init` sobre cuántas etapas debes solicitar información o, en el caso de la opción `--bootstrap`, para cuántas etapas debes crear los recursos de infraestructura. Para ver un ejemplo de archivo metadata.json que declara una canalización con dos etapas, consulta [metadata.json](#).

Proyectos de ejemplo

Estos son ejemplos de proyectos, cada uno de los cuales incluye una plantilla de Cookiecutter, un archivo `questions.json` y un archivo `metadata.json`:

- Ejemplo de Jenkins: [plantilla de canalización de Jenkins en dos etapas](#)
- CodePipeline Ejemplo: plantilla de [CodePipeline canalización en dos etapas](#)

Archivos de ejemplo

El siguiente conjunto de archivos muestra cómo se asocian las preguntas del archivo `questions.json` a las entradas del archivo de plantilla de Cookiecutter. Tenga en cuenta que estos ejemplos son fragmentos de archivos, no archivos completos. Para ver ejemplos de archivos completos, consulta primero la sección [Proyectos de ejemplo](#) de este tema.

Ejemplo: **questions.json**

```
{
  "questions": [{
    "key": "intro",
    "question": "\nThis template configures a pipeline that deploys a serverless
application to a testing and a production stage.\n",
    "kind": "info"
  }, {
    "key": "pipeline_user_jenkins_credential_id",
    "question": "What is the Jenkins credential ID (via Jenkins plugin \"aws-
credentials\") for pipeline user access key?",
    "isRequired": true
  }, {
    "key": "sam_template",
```

```
"question": "What is the template file path?",
"default": "template.yaml"
}, {
  ...
```

Ejemplo: **cookiecutter.json**

```
{
  "outputDir": "aws-sam-pipeline",
  "pipeline_user_jenkins_credential_id": "",
  "sam_template": "",
  ...
```

Ejemplo: **Jenkinsfile**

```
pipeline {
  agent any
  environment {
    PIPELINE_USER_CREDENTIAL_ID =
'{{cookiecutter.pipeline_user_jenkins_credential_id}}'
    SAM_TEMPLATE = '{{cookiecutter.sam_template}}'
    ...
```

Automatice la implementación de su AWS SAM aplicación

En AWS SAM, la forma de automatizar la implementación de su AWS SAM aplicación varía según el CI/CD sistema que utilice. Por este motivo, los ejemplos de esta sección muestran cómo configurar varios CI/CD sistemas para automatizar la creación de aplicaciones sin servidor en una imagen de contenedor de AWS SAM compilación. Estas imágenes de contenedores de compilación facilitan la creación y el empaquetado de aplicaciones sin servidor mediante el AWS SAMCLI

Los procedimientos que utiliza su CI/CD canal actual para implementar aplicaciones sin servidor AWS SAM son ligeramente diferentes según el CI/CD sistema que utilice.

En los temas siguientes se proporcionan ejemplos para configurar el CI/CD sistema para crear aplicaciones sin servidor dentro de una imagen de contenedor de AWS SAM compilación:

Temas

- [Uso AWS CodePipeline para implementar con AWS SAM](#)
- [Uso de Bitbucket Pipelines para realizar despliegues con AWS SAM](#)

- [Uso de Jenkins para implementar con AWS SAM](#)
- [Uso de GitLab CI/CD para implementar con AWS SAM](#)
- [Uso de GitHub acciones para implementar con AWS SAM](#)

Uso AWS CodePipeline para implementar con AWS SAM

Para configurar la [AWS CodePipeline](#) canalización de forma que se automatice la creación y el despliegue de AWS SAM la aplicación, la CloudFormation plantilla y el `buildspec.yml` archivo deben contener líneas que hagan lo siguiente:

1. Hacer referencia a una imagen de contenedor de compilación con el tiempo de ejecución necesario a partir de las imágenes disponibles. En el siguiente ejemplo, se utiliza la imagen del contenedor de compilación `public.ecr.aws/sam/build-nodejs20.x`.
2. Configure las etapas de canalización para ejecutar los AWS SAM comandos de la interfaz de línea de comandos (CLI) necesarios. El siguiente ejemplo ejecuta dos comandos CLI de AWS SAM: `sam build` y `sam deploy` (con las opciones necesarias).

En este ejemplo se supone que ha declarado todas las funciones y capas del archivo AWS SAM de plantilla `conruntime: nodejs20.x`.

CloudFormation fragmento de plantilla:

```
CodeBuildProject:
  Type: AWS::CodeBuild::Project
  Properties:
    Environment:
      ComputeType: BUILD_GENERAL1_SMALL
      Image: public.ecr.aws/sam/build-nodejs20.x
      Type: LINUX_CONTAINER
    ...
```

Fragmento de **`buildspec.yml`**:

```
version: 0.2
phases:
  build:
    commands:
      - sam build
```

```
- sam deploy --no-confirm-changeset --no-fail-on-empty-changeset
```

Para obtener una lista de imágenes de contenedor de compilación de Amazon Elastic Container Registry (Amazon ECR) disponibles para diferentes tiempos de ejecución, consulta [Repositorios de imágenes para AWS SAM](#).

Uso de Bitbucket Pipelines para realizar despliegues con AWS SAM

Para configurar [Bitbucket Pipeline](#) de forma que automatice la creación y el despliegue de AWS SAM la aplicación, el `bitbucket-pipelines.yml` archivo debe contener líneas que hagan lo siguiente:

1. Hacer referencia a una imagen de contenedor de compilación con el tiempo de ejecución necesario a partir de las imágenes disponibles. En el siguiente ejemplo, se utiliza la imagen del contenedor de compilación `public.ecr.aws/sam/build-nodejs20.x`.
2. Configure las etapas de canalización para ejecutar los AWS SAM comandos de la interfaz de línea de comandos (CLI) necesarios. El siguiente ejemplo ejecuta dos comandos CLI de AWS SAM: `sam build` y `sam deploy` (con las opciones necesarias).

En este ejemplo se supone que ha declarado todas las funciones y capas del archivo AWS SAM de plantilla `conruntime: nodejs20.x`.

```
image: public.ecr.aws/sam/build-nodejs20.x

pipelines:
  branches:
    main: # branch name
      - step:
          name: Build and Package
          script:
            - sam build
            - sam deploy --no-confirm-changeset --no-fail-on-empty-changeset
```

Para obtener una lista de imágenes de contenedor de compilación de Amazon Elastic Container Registry (Amazon ECR) disponibles para diferentes tiempos de ejecución, consulta [Repositorios de imágenes para AWS SAM](#).

Uso de Jenkins para implementar con AWS SAM

Para configurar tu canalización de [Jenkins](#) para automatizar la creación y el despliegue de tu AWS SAM aplicación, `Jenkinsfile` debes incluir líneas que hagan lo siguiente:

1. Hacer referencia a una imagen de contenedor de compilación con el tiempo de ejecución necesario a partir de las imágenes disponibles. En el siguiente ejemplo, se utiliza la imagen del contenedor de compilación `public.ecr.aws/sam/build-nodejs20.x`.
2. Configure las etapas de canalización para ejecutar los AWS SAM comandos de la interfaz de línea de comandos (CLI) necesarios. El siguiente ejemplo ejecuta dos comandos CLI de AWS SAM: `sam build` y `sam deploy` (con las opciones necesarias).

En este ejemplo se supone que ha declarado todas las funciones y capas del archivo AWS SAM de plantilla `conruntime: nodejs20.x`.

```
pipeline {
  agent { docker { image 'public.ecr.aws/sam/build-nodejs20.x' } }
  stages {
    stage('build') {
      steps {
        sh 'sam build'
        sh 'sam deploy --no-confirm-changeset --no-fail-on-empty-changeset'
      }
    }
  }
}
```

Para obtener una lista de imágenes de contenedor de compilación de Amazon Elastic Container Registry (Amazon ECR) disponibles para diferentes tiempos de ejecución, consulta [Repositorios de imágenes para AWS SAM](#).

Uso de GitLab CI/CD para implementar con AWS SAM

Para configurar la [GitLab](#) canalización de manera que se automatice la creación y el despliegue de AWS SAM la aplicación, el `gitlab-ci.yml` archivo debe contener líneas que hagan lo siguiente:

1. Hacer referencia a una imagen de contenedor de compilación con el tiempo de ejecución necesario a partir de las imágenes disponibles. En el siguiente ejemplo, se utiliza la imagen del contenedor de compilación `public.ecr.aws/sam/build-nodejs20.x`.
2. Configure las etapas de canalización para ejecutar los AWS SAM comandos de la interfaz de línea de comandos (CLI) necesarios. El siguiente ejemplo ejecuta dos comandos CLI de AWS SAM: `sam build` y `sam deploy` (con las opciones necesarias).

En este ejemplo se supone que ha declarado todas las funciones y capas del archivo AWS SAM de plantilla conruntime: nodejs20.x.

```
image: public.ecr.aws/sam/build-nodejs20.x
deploy:
  script:
    - sam build
    - sam deploy --no-confirm-changeset --no-fail-on-empty-changeset
```

Para obtener una lista de imágenes de contenedor de compilación de Amazon Elastic Container Registry (Amazon ECR) disponibles para diferentes tiempos de ejecución, consulta [Repositorios de imágenes para AWS SAM](#).

Uso de GitHub acciones para implementar con AWS SAM

Para configurar la [GitHub](#) canalización para automatizar la creación y el despliegue de la AWS SAM aplicación, primero debe instalar la interfaz de línea de AWS SAM comandos (CLI) en el host. Puedes usar [GitHub Actions](#) en tu GitHub flujo de trabajo para ayudarte con esta configuración.

El siguiente ejemplo de GitHub flujo de trabajo configura un host de Ubuntu mediante una serie de GitHub acciones y, a continuación, ejecuta AWS SAMCLI comandos para crear e implementar una AWS SAM aplicación:

```
on:
  push:
    branches:
      - main
jobs:
  deploy:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v3
      - uses: actions/setup-python@v3
      - uses: aws-actions/setup-sam@v2
      - uses: aws-actions/configure-aws-credentials@v1
      with:
        aws-access-key-id: ${ secrets.AWS_ACCESS_KEY_ID }
        aws-secret-access-key: ${ secrets.AWS_SECRET_ACCESS_KEY }
        aws-region: us-east-2
      - run: sam build --use-container
      - run: sam deploy --no-confirm-changeset --no-fail-on-empty-changeset
```

Para obtener una lista de imágenes de contenedor de compilación de Amazon Elastic Container Registry (Amazon ECR) disponibles para diferentes tiempos de ejecución, consulta [Repositorios de imágenes para AWS SAM](#).

Cómo utilizar la autenticación OIDC con canalizaciones AWS SAM

AWS Serverless Application Model (AWS SAM) admite la autenticación de usuarios de OpenID Connect (OIDC) para Bitbucket, GitHub Actions y la integración GitLab continua y la entrega continua (cuentas de CI/CD) platforms. With this support, you can use authorized CI/CD usuario) desde cualquiera de estas plataformas para gestionar tus canalizaciones de aplicaciones sin servidor. De lo contrario, tendrías que crear y gestionar varios usuarios AWS Identity and Access Management (IAM) para controlar el acceso a las canalizaciones. AWS SAM

Configure OIDC con Pipeline AWS SAM

Durante el proceso `aws sam pipeline bootstrap` de configuración, haz lo siguiente para configurar el OIDC con tu canalización. AWS SAM

1. Cuando se te pida que elijas un proveedor de identidad, selecciona OIDC.
2. A continuación, selecciona un proveedor de OIDC compatible.
3. Introduce la URL del proveedor del OIDC, empezando por **https://**.

Note

AWS SAM hace referencia a esta URL cuando genera el tipo de `AWS::IAM::OIDCProvider` recurso.

4. A continuación, siga las instrucciones e introduzca la información de la CI/CD plataforma necesaria para acceder a la plataforma seleccionada. Estos detalles varían según la plataforma y pueden incluir:
 - ID de cliente OIDC
 - Nombre del repositorio de código o identificador único universal (UUID)
 - Nombre del grupo o de la organización asociado con el repositorio
 - GitHub organización a la que pertenece el repositorio de código.
 - GitHub nombre del repositorio.
 - Ramificación desde la que se realizarán las implementaciones.

5. AWS SAM muestra un resumen de la configuración OIDC introducida. Introduce el número de una configuración para editarlo o pulsa Enter para continuar.
6. Cuando se te pida que confirmes la creación de los recursos necesarios para admitir la conexión OIDC introducida, pulsa Y para continuar.

AWS SAM genera un `AWS::IAM::OIDCProvider` AWS CloudFormation recurso con la configuración proporcionada que asume la función de ejecución de la canalización. Para obtener más información sobre este tipo de CloudFormation recurso, consulte [AWS::IAM::OIDCProvider](#) en la Guía del AWS CloudFormation usuario.

Note

Si el recurso del proveedor de identidad (IdP) ya existe en su cuenta Cuenta de AWS, AWS SAM haga referencia a él en lugar de crear uno nuevo.

Ejemplo

El siguiente es un ejemplo de cómo configurar el OIDC con Pipeline. AWS SAM

```
Select a permissions provider:
```

- 1 - IAM (default)
- 2 - OpenID Connect (OIDC)

```
Choice (1, 2): 2
```

```
Select an OIDC provider:
```

- 1 - GitHub Actions
- 2 - GitLab
- 3 - Bitbucket

```
Choice (1, 2, 3): 1
```

```
Enter the URL of the OIDC provider [https://token.actions.githubusercontent.com]:
```

```
Enter the OIDC client ID (sometimes called audience) [sts.amazonaws.com]:
```

```
Enter the GitHub organization that the code repository belongs to. If there is no  
organization enter your username instead: my-org
```

```
Enter GitHub repository name: testing
```

```
Enter the name of the branch that deployments will occur from [main]:
```

```
[3] Reference application build resources
```

```
Enter the pipeline execution role ARN if you have previously created one, or we will  
create one for you []:
```



```
Enter the CloudFormation execution role ARN if you have previously created one, or we
will create one for you []:
```

```
Please enter the artifact bucket ARN for your Lambda function. If you do not have a
bucket, we will create one for you []:
```

```
Does your application contain any IMAGE type Lambda functions? [y/N]:
```

[4] Summary

Below is the summary of the answers:

- 1 - Account: 123456
- 2 - Stage configuration name: dev
- 3 - Region: us-east-1
- 4 - OIDC identity provider URL: https://token.actions.githubusercontent.com
- 5 - OIDC client ID: sts.amazonaws.com
- 6 - GitHub organization: my-org
- 7 - GitHub repository: testing
- 8 - Deployment branch: main
- 9 - Pipeline execution role: [to be created]
- 10 - CloudFormation execution role: [to be created]
- 11 - Artifacts bucket: [to be created]
- 12 - ECR image repository: [skipped]

Press enter to confirm the values above, or select an item to edit the value:

This will create the following required resources for the 'dev' configuration:

- IAM OIDC Identity Provider
- Pipeline execution role
- CloudFormation execution role
- Artifact bucket

Should we proceed with the creation? [y/N]:

Más información

Para obtener más información sobre el uso de OIDC con AWS SAM Pipeline, consulte [sam pipeline bootstrap](#)

Introducción al uso sam sync sincronizar con Nube de AWS

El AWS Serverless Application Model comando Command Line Interface (AWS SAMCLI) `sam sync` ofrece opciones para sincronizar rápidamente los cambios de la aplicación local con el Nube de AWS. Utiliza `sam sync` al desarrollar sus aplicaciones para:

1. Detecta y sincroniza automáticamente los cambios locales en Nube de AWS.
2. Personalizar los cambios locales que se sincronizan con la Nube de AWS.

3. Preparar tu aplicación en la nube para probarla y validarla.

Con `sam sync`, puede crear un flujo de trabajo de desarrollo rápido que reduzca el tiempo necesario para sincronizar los cambios locales con la nube para realizar pruebas y validaciones.

Note

Se recomienda usar el comando `sam sync` para entornos de desarrollo. Para los entornos de producción, recomendamos utilizar `sam deploy` o configurar una canalización de integración y entrega continuas (CI/CD). Para obtener más información, consulte [Implemente su aplicación y sus recursos con AWS SAM](#).

El `sam sync` comando forma parte de AWS SAM Accelerate. AWS SAM Accelerate proporciona herramientas que puede utilizar para acelerar la experiencia de desarrollo y prueba de aplicaciones sin servidor en. Nube de AWS

Temas

- [Detecte y sincronice automáticamente los cambios locales en el Nube de AWS](#)
- [Personalice los cambios locales que se sincronizan con el Nube de AWS](#)
- [Preparar tu aplicación en la nube para probarla y validarla.](#)
- [Opciones para el comando de sincronización `sam`](#)
- [Resolución de problemas](#)
- [Ejemplos](#)
- [Más información](#)

Detecte y sincronice automáticamente los cambios locales en el Nube de AWS

Ejecute `sam sync` con la opción `--watch` para comenzar a sincronizar tu aplicación con la Nube de AWS. Esto hace lo siguiente:

1. Crear tu aplicación: este proceso es similar a usar el comando `sam build`.
2. Implementar tu aplicación: la CLI de AWS SAM implementa tu aplicación en AWS CloudFormation utilizando su configuración predeterminada. Se usan los siguientes valores predeterminados:

- a. AWS las credenciales y los ajustes de configuración generales que se encuentran en la carpeta `.aws` de usuario.
- b. La configuración de implementación de la aplicación se encuentra en el archivo `samconfig.toml` de la aplicación.

Si no se encuentran los valores predeterminados, la CLI de AWS SAM se lo comunicará y saldrá del proceso de sincronización.

3. Esté atento a los cambios locales: la CLI de AWS SAM sigue ejecutándose y vigila los cambios locales en tu aplicación. Esto es lo que ofrece la opción `--watch`.

Esta opción puede que esté activada de forma predeterminada. Para ver los valores predeterminados, consulta el archivo `samconfig.toml` de tu aplicación. A continuación se muestra un ejemplo de un archivo:

```
...
[default.sync]
[default.sync.parameters]
watch = true
...
```

4. Sincronizar los cambios locales con el Nube de AWS: cuando realizas cambios locales, AWS SAMCLI detecta y sincroniza esos cambios con el Nube de AWS método más rápido disponible. Según el tipo de cambio, puede ocurrir lo siguiente:
 - a. Si el recurso actualizado es compatible con las API de AWS servicio, lo AWS SAMCLI usará para implementar los cambios. Esto da como resultado una sincronización rápida para actualizar el recurso en la Nube de AWS.
 - b. Si el recurso actualizado no admite las API de AWS servicio, AWS SAMCLI realizará una CloudFormation implementación. Esto actualiza toda la aplicación en la Nube de AWS. Si bien no es tan rápido, evita que tenga que iniciar una implementación manualmente.

Dado que el `sam sync` comando actualiza automáticamente la aplicación en el Nube de AWS, solo se recomienda su uso en entornos de desarrollo. Cuando ejecuta `sam sync`, se le pedirá que confirme:

```
**The sync command should only be used against a development stack**.
```

```
Confirm that you are synchronizing a development stack.
```

```
Enter Y to proceed with the command, or enter N to cancel:  
[Y/n]: ENTER
```

Personalice los cambios locales que se sincronizan con el Nube de AWS

Proporcionar opciones para personalizar los cambios locales que se sincronizan con la Nube de AWS. Esto puede acelerar el tiempo que se tarda en ver los cambios locales en la nube para probarlos y validarlos.

Por ejemplo, ofrezca la `--code` opción de sincronizar solo los cambios de código, como el código de la función de AWS Lambda evento. Durante el desarrollo, si se centra específicamente en el código Lambda, sus cambios llegarán rápidamente a la nube para probarlos y validarlos. A continuación, se muestra un ejemplo:

```
$ sam sync --code --watch
```

Para sincronizar solo los cambios de código de una función o capa de Lambda específica, utiliza la opción `--resource-id`. A continuación, se muestra un ejemplo:

```
$ sam sync --code --resource-id HelloWorldFunction --resource-id HelloWorldLayer
```

Preparar tu aplicación en la nube para probarla y validarla.

El comando `sam sync` busca automáticamente el método más rápido disponible para actualizar la aplicación en la Nube de AWS. Esto puede acelerar sus flujos de trabajo de desarrollo y pruebas en la nube. Al utilizar las API de AWS servicio, puede desarrollar, sincronizar y probar rápidamente los recursos compatibles. Para ver un ejemplo práctico, consulte el [módulo 6: AWS SAM Accelerate](#) in The Complete AWS SAM Workshop.

Opciones para el comando de sincronización `sam`

Las siguientes son algunas de las opciones principales que puede utilizar para modificar el comando `sam sync`. Para obtener una lista de todas las opciones, consulta [sam sync](#).

Realice una sola vez CloudFormation Implementación

Usa la opción `--no-watch` para desactivar la sincronización automática. A continuación, se muestra un ejemplo:

```
$ sam sync --no-watch
```

AWS SAMCLI realizará un CloudFormation despliegue único. Este comando agrupa las acciones realizadas por los comandos `sam build` y `sam deploy`.

Omita la inicial CloudFormation Implementación

Puede personalizar si se requiere una CloudFormation implementación cada vez que `sam sync` se ejecute.

- Asegúrese `--no-skip-deploy-sync` de que se requiera una CloudFormation implementación cada vez `sam sync` que se ejecute. Esto garantiza que su infraestructura local esté sincronizada CloudFormation, lo que evita desviaciones. El uso de esta opción añade tiempo adicional a su flujo de trabajo de desarrollo y pruebas.
- Proporcione esta opción `--skip-deploy-sync` para que la CloudFormation implementación sea opcional. AWS SAMCLI comparará su AWS SAM plantilla local con la CloudFormation plantilla implementada y omitirá la CloudFormation implementación inicial si no se detecta ningún cambio. Omitir la CloudFormation implementación puede ahorrarle tiempo a la hora de sincronizar los cambios locales con la. Nube de AWS

Si no se detecta ningún cambio, AWS SAMCLI seguirá realizando un CloudFormation despliegue en los siguientes escenarios:

- Si han pasado 7 días o más desde la última CloudFormation implementación.
- Si se detecta una gran cantidad de cambios en el código de la función Lambda, la CloudFormation implementación es el método más rápido para actualizar la aplicación.

A continuación, se muestra un ejemplo:

```
$ sam sync --skip-deploy-sync
```

Sincronice un recurso de una pila anidada

Para sincronizar un recurso de una pila anidada

1. Proporcione la pila raíz utilizando `--stack-name`.
2. Identifique el recurso en la pila anidada con el siguiente formato: *nestedStackId/resourceId*.

3. Proporciona el recurso en la pila anidada usando `--resource-id`.

A continuación, se muestra un ejemplo:

```
$ sam sync --code --stack-name sam-app --resource-id myNestedStack/HelloWorldFunction
```

Para obtener más información sobre cómo crear aplicaciones anidadas, consulta [Reutilice el código y los recursos mediante aplicaciones anidadas en AWS SAM](#).

Especifique una específica CloudFormation pila para actualizar

Para especificar una CloudFormation pila específica para actualizarla, proporcione la `--stack-name` opción. A continuación, se muestra un ejemplo:

```
$ sam sync --stack-name dev-sam-app
```

Acelere los tiempos de compilación al compilar el proyecto en la carpeta de origen

Para consultar los tiempos de ejecución y los métodos de compilación compatibles, puedes usar la opción de `--build-in-source` para compilar el proyecto directamente en la carpeta de origen. De forma predeterminada, se AWS SAM CLI compila en un directorio temporal, lo que implica copiar el código fuente y los archivos del proyecto. Con `--build-in-source` las AWS SAM CLI compilaciones directamente en la carpeta de origen, lo que acelera el proceso de compilación al eliminar la necesidad de copiar los archivos a un directorio temporal.

Para obtener una lista de tiempos de ejecución y métodos de compilación compatibles, consulta [--build-in-source](#).

Especificar archivos y carpetas que no iniciarán una sincronización

Utiliza la opción `--watch-exclude` para especificar cualquier archivo o carpeta que no iniciará una sincronización cuando se actualice. Para obtener más información acerca de esta opción, consulta [--watch-exclude](#).

El siguiente es un ejemplo que excluye el archivo `package-lock.json` asociado a nuestra función de `HelloWorldFunction`:

```
$ sam sync --watch --watch-exclude HelloWorldFunction=package-lock.json
```

Cuando se ejecute este comando, AWS SAM CLI se iniciará el proceso de sincronización. Esta incluye lo siguiente:

- Ejecute `sam build` para crear las funciones y preparar la aplicación para la implementación.
- Ejecute `sam deploy` para implementar la aplicación.
- Esté atento a los cambios en la aplicación.

Cuando modifiquemos el `package-lock.json` archivo, AWS SAM CLI no se iniciará la sincronización. Cuando se actualice otro archivo, AWS SAM CLI se iniciará una sincronización, que incluirá el `package-lock.json` archivo.

El siguiente es un ejemplo de cómo especificar una función de Lambda de una pila secundaria:

```
$ sam sync --watch --watch-exclude ChildStackA/MyFunction=database.sqlite3
```

Resolución de problemas

Para solucionar problemas de AWS SAMCLI, consulte [AWS SAM CLI solución de problemas](#).

Ejemplos

Uso de `sam sync` para actualizar la aplicación Hello World

En este ejemplo, empezamos inicializando la aplicación de muestra Hello World. Para obtener más información sobre esta aplicación, consulta [Tutorial: Implemente una aplicación Hello World con AWS SAM](#).

Ejecutar `sam sync` inicia el proceso de creación e implementación.

```
$ sam sync
```

```
The SAM CLI will use the AWS Lambda, Amazon API Gateway, and AWS StepFunctions APIs to
upload your code without
performing a CloudFormation deployment. This will cause drift in your CloudFormation
stack.
```

```
**The sync command should only be used against a development stack**.
```

```
Confirm that you are synchronizing a development stack.
```

```
Enter Y to proceed with the command, or enter N to cancel:
```

```
[Y/n]:
Queued infra sync. Waiting for in progress code syncs to complete...
Starting infra sync.
Manifest file is changed (new hash: 3298f13049d19cffaa37ca931dd4d421) or dependency
  folder (.aws-sam/deps/0663e6fe-a888-4efb-b908-e2344261e9c7) is missing for
  (HelloWorldFunction), downloading dependencies and copying/building source
Building codeuri: /Users/.../Demo/sync/sam-app/hello_world runtime: python3.9 metadata:
  {} architecture: x86_64 functions: HelloWorldFunction
Running PythonPipBuilder:CleanUp
Running PythonPipBuilder:ResolveDependencies
Running PythonPipBuilder:CopySource

Build Succeeded

Successfully packaged artifacts and wrote output template to file /var/
folders/45/5ct135bx3fn2551_ptl5g6_80000gr/T/tmpx_5t4u3f.
Execute the following command to deploy the packaged template
sam deploy --template-file /var/folders/45/5ct135bx3fn2551_ptl5g6_80000gr/T/tmpx_5t4u3f
--stack-name <YOUR STACK NAME>
```

Deploying with following values

=====

```
Stack name           : sam-app
Region               : us-west-2
Disable rollback     : False
Deployment s3 bucket : aws-sam-cli-managed-default-samclisam-s3-demo-
bucket-1a4x26zbcdkqr
Capabilities          : ["CAPABILITY_NAMED_IAM", "CAPABILITY_AUTO_EXPAND"]
Parameter overrides  : {}
Signing Profiles      : null
```

Initiating deployment

=====

2023-03-17 11:17:19 - Waiting for stack create/update to complete

CloudFormation events from stack operations (refresh every 0.5 seconds)

```
-----
ResourceStatus      ResourceType
LogicalResourceId    ResourceStatusReason
-----
```


CREATE_IN_PROGRESS	AWS::CloudFormation::Stack	sam-app
Transformation succeeded		
CREATE_IN_PROGRESS	AWS::CloudFormation::Stack	
AwsSamAutoDependencyLayerNestedSt	-	
		ack
CREATE_IN_PROGRESS	AWS::IAM::Role	
HelloWorldFunctionRole	-	
CREATE_IN_PROGRESS	AWS::IAM::Role	
HelloWorldFunctionRole	Resource creation Initiated	
CREATE_IN_PROGRESS	AWS::CloudFormation::Stack	
AwsSamAutoDependencyLayerNestedSt	Resource creation Initiated	
		ack
CREATE_COMPLETE	AWS::IAM::Role	
HelloWorldFunctionRole	-	
CREATE_COMPLETE	AWS::CloudFormation::Stack	
AwsSamAutoDependencyLayerNestedSt	-	
		ack
CREATE_IN_PROGRESS	AWS::Lambda::Function	
HelloWorldFunction	-	
CREATE_IN_PROGRESS	AWS::Lambda::Function	
HelloWorldFunction	Resource creation Initiated	
CREATE_COMPLETE	AWS::Lambda::Function	
HelloWorldFunction	-	
CREATE_IN_PROGRESS	AWS::ApiGateway::RestApi	
ServerlessRestApi	-	
CREATE_IN_PROGRESS	AWS::ApiGateway::RestApi	
ServerlessRestApi	Resource creation Initiated	
CREATE_COMPLETE	AWS::ApiGateway::RestApi	
ServerlessRestApi	-	
CREATE_IN_PROGRESS	AWS::ApiGateway::Deployment	
ServerlessRestApiDeployment47fc2d	-	
		5f9d
CREATE_IN_PROGRESS	AWS::Lambda::Permission	
HelloWorldFunctionHelloWorldPermi	-	
		ssionProd
CREATE_IN_PROGRESS	AWS::Lambda::Permission	
HelloWorldFunctionHelloWorldPermi	Resource creation Initiated	
		ssionProd
CREATE_IN_PROGRESS	AWS::ApiGateway::Deployment	
ServerlessRestApiDeployment47fc2d	Resource creation Initiated	
		5f9d
CREATE_COMPLETE	AWS::ApiGateway::Deployment	
ServerlessRestApiDeployment47fc2d	-	
		5f9d

```

CREATE_IN_PROGRESS      AWS::ApiGateway::Stage
  ServerlessRestApiProdStage      -
CREATE_IN_PROGRESS      AWS::ApiGateway::Stage
  ServerlessRestApiProdStage      Resource creation Initiated
CREATE_COMPLETE          AWS::ApiGateway::Stage
  ServerlessRestApiProdStage      -
CREATE_COMPLETE          AWS::Lambda::Permission
  HelloWorldFunctionHelloWorldPermi -
                                     ssionProd
CREATE_COMPLETE          AWS::CloudFormation::Stack      sam-app
  -

```

CloudFormation outputs from deployed stack

Outputs

```

Key      HelloWorldFunctionIamRole
Description      Implicit IAM Role created for Hello World function
Value      arn:aws:iam::012345678910:role/sam-app-HelloWorldFunctionRole-
BUFVM02PJYF

Key      HelloWorldApi
Description      API Gateway endpoint URL for Prod stage for Hello World function
Value      https://pcrx5gdaof.execute-api.us-west-2.amazonaws.com/Prod/hello/

Key      HelloWorldFunction
Description      Hello World Lambda Function ARN
Value      arn:aws:lambda:us-west-2:012345678910:function:sam-app-
HelloWorldFunction-2PlN6TPTQoco

```

Stack creation succeeded. Sync infra completed.

Infra sync completed.

CodeTrigger not created as CodeUri or DefinitionUri is missing for ServerlessRestApi.

Una vez completada la implementación, modificamos el código HelloWorldFunction. AWS SAMCLIDetecta este cambio y sincroniza nuestra aplicación con. Nube de AWS Como AWS Lambda es compatible con las API de AWS servicio, se realiza una sincronización rápida.

Syncing Lambda Function HelloWorldFunction...

Manifest is not changed for (HelloWorldFunction), running incremental build

```
Building codeuri: /Users/.../Demo/sync/sam-app/hello_world runtime: python3.9 metadata:
  {} architecture: x86_64 functions: HelloWorldFunction
Running PythonPipBuilder:CopySource
Finished syncing Lambda Function HelloWorldFunction.
```

A continuación, modificamos el punto final de nuestra API en la AWS SAM plantilla de la aplicación. Cambiamos `/hello` por `/helloworld`.

```
AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31
...
Resources:
  HelloWorldFunction:
    ...
    Properties:
      ...
      Events:
        HelloWorld:
          Type: Api
          Properties:
            Path: /helloworld
            Method: get
```

Como el recurso Amazon API Gateway no es compatible con la API de AWS servicio, realiza una CloudFormation implementación AWS SAMCLI automáticamente. El siguiente es un ejemplo de output:

```
Queued infra sync. Waiting for in progress code syncs to complete...
Starting infra sync.
Manifest is not changed for (HelloWorldFunction), running incremental build
Building codeuri: /Users/.../Demo/sync/sam-app/hello_world runtime: python3.9 metadata:
  {} architecture: x86_64 functions: HelloWorldFunction
Running PythonPipBuilder:CopySource

Build Succeeded

Successfully packaged artifacts and wrote output template to file /var/
folders/45/5ct135bx3fn2551_pt15g6_80000gr/T/tmpuabo0jb9.
Execute the following command to deploy the packaged template
sam deploy --template-file /var/folders/45/5ct135bx3fn2551_pt15g6_80000gr/T/tmpuabo0jb9
--stack-name <YOUR STACK NAME>
```

Deploying with following values

=====

```
Stack name           : sam-app
Region               : us-west-2
Disable rollback     : False
Deployment s3 bucket : aws-sam-cli-managed-default-samclisam-s3-demo-
bucket-1a4x26zbcdkqr
Capabilities          : ["CAPABILITY_NAMED_IAM", "CAPABILITY_AUTO_EXPAND"]
Parameter overrides  : {}
Signing Profiles      : null
```

Initiating deployment

=====

2023-03-17 14:41:18 - Waiting for stack create/update to complete

CloudFormation events from stack operations (refresh every 0.5 seconds)

ResourceStatus	LogicalResourceId	ResourceType	ResourceStatusReason
UPDATE_IN_PROGRESS	Transformation	AWS::CloudFormation::Stack	sam-app succeeded
UPDATE_IN_PROGRESS	AwsSamAutoDependencyLayerNestedSt	AWS::CloudFormation::Stack	-
UPDATE_COMPLETE	AwsSamAutoDependencyLayerNestedSt	AWS::CloudFormation::Stack	-
UPDATE_IN_PROGRESS	ServerlessRestApi	AWS::ApiGateway::RestApi	-
UPDATE_COMPLETE	ServerlessRestApi	AWS::ApiGateway::RestApi	-
CREATE_IN_PROGRESS	ServerlessRestApiDeployment8cf30e	AWS::ApiGateway::Deployment	-
UPDATE_IN_PROGRESS	HelloWorldFunctionHelloWorldPermi	AWS::Lambda::Permission	Requested update requires the creation of a new physical resource; hence creating one.

UPDATE_IN_PROGRESS	AWS::Lambda::Permission	
HelloWorldFunctionHelloWorldPermi	Resource creation Initiated	ssionProd
CREATE_IN_PROGRESS	AWS::ApiGateway::Deployment	
ServerlessRestApiDeployment8cf30e	Resource creation Initiated	d3cd
CREATE_COMPLETE	AWS::ApiGateway::Deployment	
ServerlessRestApiDeployment8cf30e	-	d3cd
UPDATE_IN_PROGRESS	AWS::ApiGateway::Stage	
ServerlessRestApiProdStage	-	
UPDATE_COMPLETE	AWS::ApiGateway::Stage	
ServerlessRestApiProdStage	-	
UPDATE_COMPLETE	AWS::Lambda::Permission	
HelloWorldFunctionHelloWorldPermi	-	ssionProd
UPDATE_COMPLETE_CLEANUP_IN_PROGRE	AWS::CloudFormation::Stack	ssionProd
-		sam-app
SS		
DELETE_IN_PROGRESS	AWS::Lambda::Permission	
HelloWorldFunctionHelloWorldPermi	-	ssionProd
DELETE_IN_PROGRESS	AWS::ApiGateway::Deployment	
ServerlessRestApiDeployment47fc2d	-	5f9d
DELETE_COMPLETE	AWS::ApiGateway::Deployment	
ServerlessRestApiDeployment47fc2d	-	5f9d
UPDATE_COMPLETE	AWS::CloudFormation::Stack	
AwsSamAutoDependencyLayerNestedSt	-	ack
DELETE_COMPLETE	AWS::Lambda::Permission	
HelloWorldFunctionHelloWorldPermi	-	ssionProd
UPDATE_COMPLETE	AWS::CloudFormation::Stack	ssionProd
-		sam-app

CloudFormation outputs from deployed stack		

Outputs		

Key	HelloWorldFunctionIamRole	
Description	Implicit IAM Role created for Hello World function	

```
Value      arn:aws:iam::012345678910:role/sam-app-HelloWorldFunctionRole-
BUFVM02PJIYF

Key        HelloWorldApi
Description API Gateway endpoint URL for Prod stage for Hello World function
Value      https://pcrx5gdaof.execute-api.us-west-2.amazonaws.com/Prod/hello/

Key        HelloWorldFunction
Description Hello World Lambda Function ARN
Value      arn:aws:lambda:us-west-2:012345678910:function:sam-app-
HelloWorldFunction-2PlN6TPTQoco
```

Stack update succeeded. Sync infra completed.

Infra sync completed.

Más información

Para ver una descripción de las opciones de `sam sync`, consulta [sam sync](#).

Supervise su aplicación sin servidor con AWS SAM

Tras implementar la aplicación sin servidor, puede supervisarla para obtener información sobre sus operaciones y detectar anomalías, lo que puede ayudar a solucionar problemas. En esta sección se proporcionan detalles sobre la supervisión de una aplicación sin servidor. Esto incluye información sobre cómo configurar Amazon CloudWatch para que te notifique cuando detecte anomalías. También proporciona información sobre cómo trabajar con los registros, incluyendo información sobre los errores y consejos para ver, filtrar, buscar y guardar los registros.

Temas

- [Uso de CloudWatch Application Insights para monitorear sus aplicaciones AWS SAM sin servidor](#)
- [Trabajar con inicios de sesión AWS SAM](#)

Uso de CloudWatch Application Insights para monitorear sus aplicaciones AWS SAM sin servidor

Amazon CloudWatch Application Insights le ayuda a supervisar los AWS recursos de sus aplicaciones para identificar posibles problemas. Puede analizar los datos AWS de los recursos en busca de señales de problemas y crear paneles automatizados para visualizarlos. Puede configurar CloudWatch Application Insights para utilizarla con sus AWS Serverless Application Model (AWS SAM) aplicaciones. Para obtener más información sobre CloudWatch Application Insights, consulte [Amazon CloudWatch Application Insights](#) en la Guía del CloudWatch usuario de Amazon.

Temas

- [Configurar CloudWatch Application Insights con AWS SAM](#)
- [Pasos a seguir a continuación](#)

Configurar CloudWatch Application Insights con AWS SAM

Configure CloudWatch Application Insights para sus AWS SAM aplicaciones mediante la interfaz de línea de AWS SAM comandos (AWS SAMCLI) o mediante sus AWS SAM plantillas.

Configura mediante la CLI de AWS SAM

Al inicializar la aplicación con `sam init`, active CloudWatch Application Insights mediante el flujo interactivo o mediante la `--application-insights` opción.

Para activar CloudWatch Application Insights a través del flujo AWS SAMCLI interactivo, introdúzcalo **y** cuando se le solicite.

```
Would you like to enable monitoring using CloudWatch Application Insights?  
For more info, please view https://docs.aws.amazon.com/AmazonCloudWatch/latest/  
monitoring/cloudwatch-application-insights.html [y/N]:
```

Para activar CloudWatch Application Insights con la `--application-insights` opción, haga lo siguiente.

```
sam init --application-insights
```

Para obtener más información sobre el uso del comando `sam init`, consulta [sam init](#).

Configure mediante AWS SAM plantillas

Active CloudWatch Application Insights definiendo los

`AWS::ApplicationInsights::Application` recursos `AWS::ResourceGroups::Group` y los recursos de sus AWS SAM plantillas.

```
AWSTemplateFormatVersion: '2010-09-09'  
Transform: AWS::Serverless-2016-10-31  
...  
Resources:  
  ApplicationResourceGroup:  
    Type: AWS::ResourceGroups::Group  
    Properties:  
      Name:  
        Fn::Join:  
          - ''  
          - - ApplicationInsights-SAM-  
            - Ref: AWS::StackName  
      ResourceQuery:  
        Type: CLOUDFORMATION_STACK_1_0  
  ApplicationInsightsMonitoring:  
    Type: AWS::ApplicationInsights::Application  
    Properties:  
      ResourceGroupName:
```



```

Fn::Join:
- ''
- - ApplicationInsights-SAM-
  - Ref: AWS::StackName
  AutoConfigurationEnabled: 'true'
  DependsOn: ApplicationResourceGroup

```

- **AWS::ResourceGroups::Group**— Crea un grupo para organizar sus AWS recursos con el fin de gestionar y automatizar las tareas de una gran cantidad de recursos a la vez. Aquí puede crear un grupo de recursos para usarlo con CloudWatch Application Insights. Para obtener más información sobre este tipo de recurso, consulta [AWS::ResourceGroups::Group](#) en la Guía del usuario AWS CloudFormation .
- **AWS::ApplicationInsights::Application**— Configura CloudWatch Application Insights para el grupo de recursos. Para obtener más información sobre este tipo de recurso, consulta [AWS::ApplicationInsights::Application](#) en la Guía del usuario AWS CloudFormation .

Ambos recursos se transfieren automáticamente CloudFormation al implementar la aplicación. Puede usar la CloudFormation sintaxis de la AWS SAM plantilla para configurar aún más CloudWatch Application Insights. Para obtener más información, consulta [Uso de CloudFormation plantillas](#) en la Guía del CloudWatch usuario de Amazon.

Al utilizar el `sam init --application-insights` comando, ambos recursos se generan automáticamente en la AWS SAM plantilla. A continuación, se muestra un ejemplo de una plantilla generada.

```

AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31
Description: >
  sam-app-test

  Sample SAM Template for sam-app-test

# More info about Globals: https://github.com/awslabs/serverless-application-model/
blob/master/docs/globals.rst
Globals:
  Function:
    Timeout: 3
    MemorySize: 128

Resources:
  HelloWorldFunction:

```

```

    Type: AWS::Serverless::Function # More info about Function Resource:
    https://github.com/awslabs/serverless-application-model/blob/master/
versions/2016-10-31.md#awsserverlessfunction
    Properties:
      CodeUri: hello_world/
      Handler: app.lambda_handler
      Runtime: python3.9
      Architectures:
      - x86_64
      Events:
        HelloWorld:
          Type: Api # More info about API Event Source: https://github.com/awslabs/
serverless-application-model/blob/master/versions/2016-10-31.md#api
          Properties:
            Path: /hello
            Method: get

ApplicationResourceGroup:
  Type: AWS::ResourceGroups::Group
  Properties:
    Name:
      Fn::Join:
      - ''
      - - ApplicationInsights-SAM-
        - Ref: AWS::StackName
    ResourceQuery:
      Type: CLOUDFORMATION_STACK_1_0
ApplicationInsightsMonitoring:
  Type: AWS::ApplicationInsights::Application
  Properties:
    ResourceGroupName:
      Fn::Join:
      - ''
      - - ApplicationInsights-SAM-
        - Ref: AWS::StackName
    AutoConfigurationEnabled: 'true'
    DependsOn: ApplicationResourceGroup

Outputs:
  # ServerlessRestApi is an implicit API created out of Events key under
  Serverless::Function
  # Find out more about other implicit resources you can reference within SAM
  # https://github.com/awslabs/serverless-application-model/blob/master/docs/internals/
  generated_resources.rst#api

```

```
HelloWorldApi:
  Description: API Gateway endpoint URL for Prod stage for Hello World function
  Value: !Sub "https://${ServerlessRestApi}.execute-api.${AWS::Region}.amazonaws.com/Prod/hello/"
HelloWorldFunction:
  Description: Hello World Lambda Function ARN
  Value: !GetAtt HelloWorldFunction.Arn
HelloWorldFunctionIamRole:
  Description: Implicit IAM Role created for Hello World function
  Value: !GetAtt HelloWorldFunctionRole.Arn
```

Pasos a seguir a continuación

Tras configurar CloudWatch Application Insights, utilícelo `sam build` `sam deploy` para crear y desplegar la aplicación. Todos los recursos compatibles con CloudWatch Application Insights se configurarán para su supervisión.

- Para ver una lista de los recursos compatibles, consulta [Registros y métricas compatibles](#) en la Guía del CloudWatch usuario de Amazon.
- Para obtener información sobre cómo acceder a CloudWatch Application Insights, consulte [Access CloudWatch Application Insights](#) en la Guía del CloudWatch usuario de Amazon.

Trabajar con inicios de sesión AWS SAM

Para simplificar la solución de problemas, la CLI de AWS SAM tiene un comando llamado [registros de spam](#). Este comando le permite recuperar los registros generados por la función de eventos de Lambda desde la línea de comandos.

Note

El `sam logs` comando funciona para todas las AWS Lambda funciones, no solo para las que se despliegan. AWS SAM

Obtención de registros mediante CloudFormation pila

Cuando tu función forma parte de una CloudFormation pila, puedes recuperar los registros mediante el identificador lógico de la función:

```
sam logs -n HelloWorldFunction --stack-name mystack
```

Obtención de registros por nombre de función de eventos de Lambda

O bien, puede obtener registros utilizando el nombre de la función:

```
sam logs -n mystack-HelloWorldFunction-1FJ8PD
```

Rastreo de registros

Agrega la opción `--tail` para esperar a nuevos registros y verlos a medida que llegan. Esto es muy útil durante la implementación o al solucionar un problema de producción:

```
sam logs -n HelloWorldFunction --stack-name mystack --tail
```

Visualización de registros para un intervalo de tiempo específico

Puedes ver los registros de un intervalo de tiempo específico mediante las opciones `-s` y `-e`:

```
sam logs -n HelloWorldFunction --stack-name mystack -s '10min ago' -e '2min ago'
```

Filtrado de registros

Utiliza la opción `--filter` para encontrar rápidamente registros que coincidan con términos, frases o valores de sus eventos de registro:

```
sam logs -n HelloWorldFunction --stack-name mystack --filter "error"
```

En la salida, la CLI de AWS SAM subraya todas las veces que aparece la palabra "error" para que pueda localizar fácilmente la palabra clave de filtro en la salida del registro.

Resaltado de errores

Cuando la función de Lambda se bloquea o se agota el tiempo de espera, la CLI de AWS SAM resalta el mensaje de tiempo de espera en rojo. Esto ayuda a localizar fácilmente las ejecuciones específicas cuyo tiempo de espera se ha agotado en un flujo gigante de salida de registro.

Impresión clara de código JSON

Si sus mensajes de registro imprimen cadenas JSON, la CLI de AWS SAM imprime automáticamente el código JSON para ayudarle a analizarlo y entenderlo visualmente.

AWS SAM Referencia de

Esta sección contiene material AWS SAM de referencia. Incluye material de referencia de la CLI de AWS SAM, como información de referencia sobre los comandos de la CLI de AWS SAM e información adicional sobre la CLI de AWS SAM, como información sobre la configuración, el control de versiones y la solución de problemas. Además, esta sección incluye información de referencia sobre la AWS SAM especificación y la AWS SAM plantilla, como información de referencia sobre los conectores, los repositorios de imágenes y las implementaciones.

AWS SAM la especificación y el AWS SAM plantilla

La AWS SAM especificación es una especificación de código abierto bajo la licencia Apache 2.0. La versión actual de la AWS SAM especificación está disponible en. [AWS SAM plantilla](#) AWS SAM La especificación incluye una sintaxis abreviada simplificada que se utiliza para definir las funciones, los eventos, las API, las configuraciones y los permisos de su aplicación sin servidor.

La interacción con las AWS SAM especificaciones se realiza a través del directorio de proyectos de la AWS SAM aplicación, que son las carpetas y los archivos que se crean al ejecutar el `sam init` comando. Este directorio incluye la AWS SAM plantilla, un archivo importante que define AWS los recursos. La AWS SAM plantilla es una extensión de una AWS CloudFormation plantilla. Para obtener una referencia completa sobre las plantillas de CloudFormation , consulta la [Referencia de plantillas de](#) en la Guía del usuario de AWS CloudFormation .

AWS SAM CLI Referencia de los comandos

La interfaz de línea de AWS Serverless Application Model comandos (AWS SAMCLI) es una herramienta de línea de comandos que puede utilizar con AWS SAM plantillas e integraciones de terceros compatibles para crear y ejecutar sus aplicaciones sin servidor.

Puedes utilizar los comandos de la CLI de AWS SAM para desarrollar, probar e implementar sus aplicaciones sin servidor en Nube de AWS. A continuación, se presentan algunos ejemplos de comandos de la CLI de AWS SAM:

- `sam init`: si es la primera vez que utilizas la CLI de AWS SAM, puedes ejecutar el comando `sam init` sin ningún parámetro para crear una aplicación Hello World. El comando genera una AWS SAM plantilla preconfigurada y un ejemplo de código de aplicación en el idioma que elija.

- `sam local invoke` y `sam local start-api`: utiliza estos comandos para probar el código de la aplicación de forma local antes de implementarlo en Nube de AWS.
- `sam logs`— Utilice este comando para recuperar los registros que genera la función de eventos de Lambda. Esto puede ayudarlo a probar y depurar la aplicación después de haberla implementado en Nube de AWS.
- `sam package`: utiliza este comando para agrupar el código y las dependencias de la aplicación en un paquete de implementación. Necesita el paquete de implementación para cargar la aplicación en Nube de AWS.
- `sam deploy`: utiliza este comando para implementar tu aplicación sin servidor en Nube de AWS. Crea los AWS recursos y establece los permisos y otras configuraciones que se definen en la AWS SAM plantilla.

Para obtener instrucciones sobre cómo instalar el AWS SAMCLI, consulte [Instala la AWS SAM CLI](#).

AWS SAM Plantillas de política de

Con AWS SAM, puede elegir entre una lista de plantillas de políticas para limitar los permisos de su AWS Lambda función a los recursos que utiliza su aplicación. Para obtener una lista de las plantillas de políticas disponibles, consulte [Tabla de plantillas de políticas](#). Para obtener información general sobre las plantillas de políticas AWS SAM, consulte [AWS SAM Plantillas de política de](#).

Temas

- [AWS SAM plantilla](#)
- [Referencia de los comandos AWS SAM CLI](#)
- [Archivo de configuración de la CLI de AWS SAM](#)
- [AWS SAM referencia de conector](#)
- [AWS SAM Plantillas de política de](#)
- [Repositorios de imágenes para AWS SAM](#)
- [Telemetría en la AWS SAMCLI](#)
- [Configure y administre el acceso a los recursos en su AWS SAM plantilla](#)

Referencia de los comandos AWS SAM CLI

En esta sección se incluye información de referencia sobre los comandos de la CLI de AWS SAM. Se incluyen detalles sobre el uso, una lista completa de las diferentes opciones disponibles para cada comando e información adicional. Cuando proceda, la información adicional incluye detalles como los argumentos, las variables de entorno y los eventos. Para obtener más información, consulte cada comando. Para obtener instrucciones sobre cómo instalar el AWS SAMCLI, consulte [Instala la AWS SAM CLI](#).

Temas

- [compilación de sam](#)
- [sam delete](#)
- [sam deploy](#)
- [sam init](#)
- [sam list](#)
- [sam local callback](#)
- [sam local execution](#)
- [sam local generate-event](#)
- [invocar sam local](#)
- [sam local start-api](#)
- [sam local start-lambda](#)
- [registros de spam](#)
- [sam package](#)
- [sam pipeline bootstrap](#)
- [sam pipeline init](#)
- [sam publish](#)
- [sam remote callback](#)
- [sam remote execution](#)
- [invocación remota de sam](#)
- [evento de prueba remoto de sam](#)
- [sam sync](#)
- [sam traces](#)

- [sam validate](#)

compilación de sam

Esta página proporciona información de referencia para el AWS Serverless Application Model comando de la interfaz de línea de `sam build` comandos (AWS SAMCLI).

- Para obtener una introducción al AWS SAMCLI, consulte [¿Qué es el AWS SAM CLI?](#)
- Para obtener documentación sobre el uso del AWS SAMCLI `sam build` comando, consulte [Introducción a la construcción con AWS SAM](#).

El comando `sam build` prepara una aplicación para los pasos posteriores del flujo de trabajo del desarrollador, como las pruebas locales o la implementación en Nube de AWS.

De uso

```
$ sam build <arguments> <options>
```

Argumentos

Resource ID (ID de recurso)

Opcional. Indica AWS SAM que se cree un único recurso declarado en una [AWS SAM plantilla](#). Los artefactos de compilación del recurso especificado serán los únicos disponibles para los siguientes comandos del flujo de trabajo, es decir, `sam package` y `sam deploy`.

Opciones

`--base-dir`, `-s` *DIRECTORY*

Resuelve las rutas relativas al código fuente de la función o capa con respecto a este directorio. Utiliza esta opción si desea cambiar la forma en que se resuelven las rutas relativas a las carpetas de código fuente. De forma predeterminada, las rutas relativas se resuelven con respecto a la ubicación de la plantilla de AWS SAM .

Además de los recursos de la aplicación o pila raíz que está creando, esta opción también se aplica a las aplicaciones o pilas anidadas.

Esta opción se aplica a los siguientes tipos y propiedades de recursos:

- Tipo de recurso: `AWS::Serverless::Function` Propiedad: `CodeUri`
- Tipo de recurso: `AWS::Serverless::Function` Atributo de recurso: `Metadata Entrada: DockerContext`
- Tipo de recurso: `AWS::Serverless::LayerVersion` Propiedad: `ContentUri`
- Tipo de recurso: `AWS::Lambda::Function` Propiedad: `Code`
- Tipo de recurso: `AWS::Lambda::LayerVersion` Propiedad: `Content`

`--beta-features` | `--no-beta-features`

Permitir o denegar las características beta.

`--build-dir`, `-b` *DIRECTORY*

La ruta a un directorio en el que se almacenan los artefactos creados. Este directorio y todo su contenido se eliminan con esta opción.

`--build-image` *TEXT*

URI de la imagen del contenedor que quiere extraer para la compilación. De forma predeterminada, AWS SAM extrae la imagen del contenedor de Amazon ECR Public. Utiliza esta opción para extraer la imagen desde otra ubicación.

Puedes especificar esta opción varias veces. Cada instancia de esta opción puede tomar una cadena o un par clave-valor. Si especificas una cadena, será el URI de la imagen del contenedor que se usará para todos los recursos de la aplicación. Por ejemplo, `sam build --use-container --build-image amazon/aws-sam-cli-build-image-python3.8`. Si especificas un par clave-valor, la clave es el nombre del recurso y el valor es el URI de la imagen del contenedor que se va a usar para ese recurso. Por ejemplo, `sam build --use-container --build-image Function1=amazon/aws-sam-cli-build-image-python3.8`. Con los pares clave-valor, puede especificar distintas imágenes del contenedor para distintos recursos.

Esta opción solo se aplica si se especifica la opción `--use-container`; de lo contrario, se producirá un error.

`--build-in-source` | `--no-build-in-source`

Proporciona `--build-in-source` para compilar el proyecto directamente en la carpeta de origen.

La opción `--build-in-source` admite los siguientes tiempos de ejecución y métodos de compilación:

- Tiempos de ejecución: cualquier tiempo de ejecución Node.js compatible con la opción [sam init --runtime](#).
- Métodos de compilación: Makefile, esbuild.

La opción `--build-in-source` no es compatible con las siguientes opciones:

- `--hook-name`
- `--use-container`

Valor predeterminado: `--no-build-in-source`

`--cached` | `--no-cached`

Habilitar o desactivar las compilaciones en caché. Usa esta opción para reutilizar artefactos de compilación que no hayan cambiado con respecto a compilaciones anteriores. AWS SAM evalúa si has cambiado algún archivo del directorio de tu proyecto. De forma predeterminada, las compilaciones no se almacenan en caché. Si se invoca la opción `--no-cached`, anula la configuración `cached = true` de `samconfig.toml`.

 Note

AWS SAM no evalúa si ha cambiado los módulos de terceros de los que depende su proyecto si no ha proporcionado una versión específica. Por ejemplo, si tu función de Python incluye un `requirements.txt` archivo con la entrada `requests=1.x` y la última versión del módulo de solicitud cambia de 1.1 a 1.2, AWS SAM no extraerá la última versión hasta que ejecutes una compilación que no esté almacenada en caché.

`--cache-dir`

El directorio en el que se almacenan los artefactos de la caché cuando se especifica `--cached`. El directorio predeterminado de la caché es `.aws-sam/cache`.

`--config-env` **TEXT**

El nombre del entorno que especifica los valores de los parámetros predeterminados del archivo de configuración que se va a utilizar. El valor por defecto es «predeterminado». Para obtener más información sobre los archivos de configuración, consulta [Archivo de configuración de la CLI de AWS SAM](#).

--config-file *PATH*

La ruta y el nombre del archivo de configuración que contiene los valores de parámetros predeterminados que se van a utilizar. El valor por defecto es «samconfig.toml» en la raíz del directorio del proyecto. Para obtener más información sobre los archivos de configuración, consulta [Archivo de configuración de la CLI de AWS SAM](#).

--container-env-var, -e *TEXT*

Las variables de entorno que transferir al contenedor de compilaciones. Puedes especificar esta opción varias veces. Cada instancia de esta opción utiliza un par clave-valor, donde la clave es la variable de recurso y entorno, y el valor es el valor de la variable de entorno. Por ejemplo:

```
--container-env-var Function1.GITHUB_TOKEN=TOKEN1 --container-env-var  
Function2.GITHUB_TOKEN=TOKEN2.
```

Esta opción solo se aplica si se especifica la opción `--use-container`; de lo contrario, se producirá un error.

--container-env-var-file, -ef *PATH*

La ruta y el nombre de un archivo JSON que contiene valores para las variables de entorno del contenedor. Para obtener más información sobre archivos de variable de entorno del contenedor, consulta [Archivo de variables del entorno del contenedor](#).

Esta opción solo se aplica si se especifica la opción `--use-container`; de lo contrario, se producirá un error.

--debug

Activa el registro de depuración para imprimir los mensajes de depuración que genera la CLI de AWS SAM y para mostrar las marcas temporales.

--docker-network *TEXT*

Especifica el nombre o el ID de una red de Docker existente a la que deben conectarse los contenedores de Docker, junto con la red en modo puente predeterminada. Si no se especifica, los contenedores de Lambda se conectan únicamente a la red Docker en modo puente predeterminada.

--exclude, -x

El nombre de los recursos que se van a excluir de `sam build`. Por ejemplo, si tu plantilla contiene `Function1`, `Function2` y `Function3` y usted ejecuta `sam build --exclude Function2`, solo se compilarán `Function1` y `Function3`.

`--help`

Muestra este mensaje y sale.

`--hook-name` *TEXT*

El nombre del enlace que se utiliza para ampliar la funcionalidad de la CLI de AWS SAM.

Valores aceptados: terraform.

`--manifest` , `-m` *PATH*

La ruta a un archivo de manifiesto de dependencias personalizado (por ejemplo, package.json) para usarlo en lugar del predeterminado.

`--mount-symlinks`

Garantiza que la CLI de AWS SAM siempre monte los enlaces simbólicos que están presentes en los archivos para compilarlos o invocarlos. Esto solo se aplica a los enlaces simbólicos del directorio de nivel superior (es decir, los que se encuentran directamente en la raíz de la función). Por defecto, los enlaces simbólicos no se montan, excepto los necesarios para utilizar `build-in-source` para `node_modules` en NodeJS.

`--no-use-container`

Una opción que permite usar el kit de herramientas de IDE para establecer el comportamiento predeterminado. También puede utilizar `sam build --no-use-container` para ejecutar una compilación en su máquina local en lugar de en un contenedor de Docker.

`--output` *[text | json]*

El formato del resultado del comando:

- `text`— Imprime un resultado normal y legible por humanos. Este es el valor predeterminado.
- `json`— Imprime resultados estructurados y legibles por máquina. Utilice este formato cuando desee consumir el resultado mediante programación, por ejemplo, en CI/CD canalizaciones, extensiones de IDE y herramientas. AI-assisted

Cuando lo especificas `--output json`, AWS SAMCLI imprime un único objeto JSON una vez finalizado el comando. En caso de éxito, este objeto incluye campos como el estado de compilación, el directorio de compilación, el archivo de plantilla y una lista de los recursos que se crearon. En caso de error, incluye un `error` objeto que describe lo que ha fallado.

--parallel

Habilitar compilaciones paralelas. Usa esta opción para crear las funciones y capas de tu AWS SAM plantilla en paralelo. De forma predeterminada, las funciones y las capas se crean en secuencia.

--parameter-overrides

(Opcional) Una cadena que contiene anulaciones de CloudFormation parámetros codificadas como pares clave-valor. Usa el mismo formato que (). AWS Command Line Interface AWS CLI Por ejemplo: 'ParameterKey=KeyPairName, ParameterValue=MyKey ParameterKey=InstanceType, ParameterValue=t1.micro'. Esta opción no es compatible con --hook-name.

--profile *TEXT*

El perfil específico de su archivo de credenciales que obtiene AWS las credenciales.

--region *TEXT*

El lugar en el Región de AWS que realizar el despliegue. Por ejemplo, us-east-1.

--save-params

Guarde los parámetros que proporciona en la línea de comandos en el archivo AWS SAM de configuración.

--skip-prepare-infra

Omite la fase de preparación si no se ha realizado ningún cambio en la infraestructura. Ejecuta con la opción --hook-name.

--skip-pull-image

Especifica si el comando debe omitir la extracción de la última imagen de Docker para el tiempo de ejecución de Lambda.

--template-file, --template, -t *PATH*

La ruta y el nombre del archivo de AWS SAM plantilla[default: template.[yaml|yml]]. Esta opción no es compatible con --hook-name.

--terraform-project-root-path

La ruta relativa o absoluta al directorio de nivel superior que contiene los archivos de configuración de Terraform o el código fuente de la función. Si estos archivos se encuentran fuera

del directorio que contiene el módulo raíz de Terraform, utilice esta opción para especificar su ruta absoluta o relativa. Esta opción requiere que `--hook-name` se establezca en `terraform`.

`--use-buildkit` | `--no-use-buildkit`

Activa o desactiva el uso del BuildKit complemento para la creación de imágenes de contenedores. Esto usará el Finch o DockerCLIs. De forma predeterminada, las compilaciones utilizan el generador heredado del SDK. Si se invoca la opción `--no-use-buildkit`, anula la configuración `use-buildkit = true` de `samconfig.toml`.

`--use-container`, `-u`

Si tus funciones dependen de paquetes que tienen dependencias compiladas de forma nativa, usa esta opción para crear tu función dentro de un Lambda-like contenedor Docker.

Ejemplo

Para ver un ejemplo y un tutorial detallados sobre el uso del subcomando `sam build`, consulte [Introducción a la construcción con AWS SAM](#).

sam delete

Esta página proporciona información de referencia para el AWS Serverless Application Model comando Command Line Interface (AWS SAMCLI) `sam delete`.

Para obtener una introducción al AWS SAMCLI, consulte [¿Qué es el AWS SAM CLI?](#)

El `sam delete` comando elimina una AWS SAM aplicación eliminando la CloudFormation pila, los artefactos que se empaquetaron e implementaron en Amazon S3 y Amazon ECR y el archivo de AWS SAM plantilla.

Este comando también comprueba si se ha implementado una pila complementaria de Amazon ECR y, de ser así, solicita al usuario que elimine esa pila y los repositorios de Amazon ECR. Si se especifica `--no-prompts`, las pilas complementarias y los repositorios de Amazon ECR se eliminan de forma predeterminada.

De uso

```
$ sam delete <options>
```

Opciones

`--config-env` *TEXT*

El nombre del entorno que especifica los valores de los parámetros predeterminados del archivo de configuración que se va a utilizar. El valor predeterminado es `default`. Para obtener más información sobre los archivos de configuración, consulta [Archivo de configuración de la CLI de AWS SAM](#).

`--config-file` *PATH*

La ruta y el nombre del archivo de configuración que contiene los valores de parámetros predeterminados que se van a utilizar. El valor por defecto es `samconfig.toml` en la raíz del directorio del proyecto. Para obtener más información sobre los archivos de configuración, consulta [Archivo de configuración de la CLI de AWS SAM](#).

`--debug`

Activa el registro de depuración para imprimir el mensaje de depuración que AWS SAMCLI genera y para mostrar las marcas temporales.

`--help`

Muestra este mensaje y sale.

`--no-prompts`

Especifique esta opción para que AWS SAM funcione en modo no interactivo. El nombre de la pila debe proporcionarse junto con la opción `--stack-name` o en el archivo de configuración `toml`.

`--profile` *TEXT*

El perfil específico del archivo de credenciales que recibe AWS las credenciales.

`--region` *TEXT*

La AWS región en la que se va a realizar el despliegue. Por ejemplo, `us-east-1`.

`--s3-bucket`

La ruta del bucket de Amazon S3 que desea eliminar.

`--s3-prefix`

El prefijo del bucket de Amazon S3 que desea eliminar.

`--save-params`

Guarde los parámetros que proporcione en la línea de comandos en el archivo AWS SAM de configuración.

`--stack-name` *TEXT*

El nombre de la CloudFormation pila que desea eliminar.

Ejemplos

El siguiente comando elimina la pila MY-STACK.

```
$ sam delete --stack-name MY-STACK
```

El siguiente comando elimina la pila MY-STACK y el bucket de S3 sam-s3-demo-bucket:

```
$ sam delete \  
  --stack-name MyStack \  
  --s3-bucket MySAMBucket
```

sam deploy

Esta página proporciona información de referencia para el AWS Serverless Application Model comando de la interfaz de línea de `sam deploy` comandos (AWS SAMCLI).

- Para obtener una introducción al AWS SAMCLI, consulte [¿Qué es el AWS SAM CLI?](#)
- Para obtener documentación sobre el uso del AWS SAMCLI `sam deploy` comando, consulte [Introducción a la implementación con AWS SAM](#).

El `sam deploy` comando implementa una aplicación para el usuario Nube de AWS . AWS CloudFormation

De uso

```
$ <environment variables> sam deploy <options>
```

Variables de entorno

SAM_CLI_POLL_DELAY

Configure la variable de SAM_CLI_POLL_DELAY entorno con un valor de segundos en su shell para configurar la frecuencia con la que la CLI de AWS SAM comprueba el estado de la CloudFormation pila, lo que resulta útil cuando se ve una limitación desde CloudFormation. Esta variable de entorno se utiliza para sondear las llamadas a la API `describe_stack`, que se realizan durante la ejecución de `sam deploy`.

A continuación se muestra un ejemplo de la variable.

```
$ SAM_CLI_POLL_DELAY=5 sam deploy
```

Opciones

--capabilities *LIST*

Una lista de capacidades que debe especificar para CloudFormation permitir la creación de determinadas pilas. Algunas plantillas de pila pueden incluir recursos que afectan a tus permisos Cuenta de AWS, por ejemplo, al crear nuevos usuarios AWS Identity and Access Management (IAM). Para estas pilas, debe reconocer explícitamente sus capacidades especificando esta opción. Los únicos valores válidos son `CAPABILITY_IAM` y `CAPABILITY_NAMED_IAM`. Si tiene recursos de IAM, puede especificar cualquiera de las dos capacidades. Si tiene recursos de IAM con nombres personalizados, debe especificar `CAPABILITY_NAMED_IAM`. Si no especifica esta opción, la operación devuelve un error `InsufficientCapabilities`.

Cuando implementa una aplicación que contiene aplicaciones anidadas, debe usar `CAPABILITY_AUTO_EXPAND` para reconocer que la aplicación contiene aplicaciones anidadas. Para obtener más información, consulte [Implementar aplicaciones anidadas](#).

--config-env *TEXT*

El nombre del entorno que especifica los valores de los parámetros predeterminados del archivo de configuración que se va a utilizar. El valor predeterminado es `default`. Para obtener más información sobre los archivos de configuración, consulta [Archivo de configuración de la CLI de AWS SAM](#).

`--config-file` *PATH*

La ruta y el nombre del archivo de configuración que contiene los valores de parámetros predeterminados que se van a utilizar. El valor por defecto es `samconfig.toml` en la raíz del directorio del proyecto. Para obtener más información sobre los archivos de configuración, consulta [Archivo de configuración de la CLI de AWS SAM](#).

`--confirm-changeset` | `--no-confirm-changeset`

Solicita confirmar si la CLI de AWS SAM implementa el conjunto de cambios calculado.

`--debug`

Active el registro de depuración para imprimir el mensaje de depuración que la CLI de AWS SAM genera y para mostrar las marcas temporales.

`--disable-rollback` | `--no-disable-rollback`

Especifica si deseas deshacer tu CloudFormation pila en caso de que se produzca un error durante una implementación. De forma predeterminada, si se produce un error durante una implementación, la CloudFormation pila vuelve al último estado estable. Si especificas `--disable-rollback` y se produce un error durante una implementación, los recursos que se crearon o actualizaron antes de que se produjera el error no se revertirán.

`--express` | `--no-express`

Especifica si se debe usar el modo CloudFormation Express. Cuando están activas, las operaciones de pila se completan una vez que se aplica la configuración de recursos. Los recursos siguen preparándose en segundo plano. El tiempo de implementación se puede reducir hasta 4 veces.

AWS SAMCLITransmite el comportamiento de reversión al CloudFormation DeploymentConfig parámetro cuando también lo especificas. `--disable-rollback`

El valor predeterminado es `--no-express`.

`--fail-on-empty-changeset` | `--no-fail-on-empty-changeset`

Especifica si desea devolver un código de salida distinto de cero si no hay cambios que realizar en la pila. El comportamiento predeterminado es devolver un código de salida distinto de cero.

`--force-upload`

Especifica esta opción para cargar artefactos incluso si coinciden con los artefactos existentes en el bucket de Amazon S3. Los artefactos coincidentes se sobrescriben.

--guided, -g

Especifica esta opción para que la CLI de AWS SAM utilice indicaciones que le guíen durante la implementación.

--help

Muestre este mensaje y salga de él.

--image-repositories *TEXT*

Un mapeo de funciones a su URI del repositorio de Amazon ECR. Haga referencia a las funciones por su ID lógico. A continuación, se muestra un ejemplo:

```
$ sam deploy --image-repositories Function1=123456789012.dkr.ecr.us-east-1.amazonaws.com/my-repo
```

Puede especificar esta opción varias veces con un solo comando.

--image-repository *TEXT*

El nombre del repositorio de Amazon ECR en el que este comando carga la imagen de la función. Esta opción es necesaria para las funciones declaradas con el tipo de paquete Image.

--kms-key-id *TEXT*

El ID de una clave AWS Key Management Service (AWS KMS) utilizada para cifrar los artefactos que están en reposo en el bucket de Amazon S3. Si no especificas esta opción, AWS SAM utiliza las claves de S3-managed cifrado de Amazon.

--metadata

Un mapa de metadatos para adjuntar a todos los artefactos a los que se hace referencia en la plantilla.

--no-execute-changeset

Indica si se debe aplicar el conjunto de cambios. Especifica esta opción si desea ver los cambios de la pila antes de aplicar el conjunto de cambios. Este comando crea un conjunto de cambios CloudFormation y, a continuación, se cierra sin aplicar el conjunto de cambios. Para aplicar el conjunto de cambios, ejecuta el mismo comando sin esta opción.

--no-progressbar

No muestre una barra de progreso al cargar artefactos en Amazon S3.

`--notification-arns` *LIST*

Una lista de los ARN temáticos de Amazon Simple Notification Service (Amazon SNS) CloudFormation asociados a la pila.

`--on-failure` [ROLLBACK | DELETE | DO_NOTHING]

Especifica la acción que se debe realizar cuando no se pueda crear una pila.

Están disponibles las siguientes opciones:

- **ROLLBACK:** hace retroceder la pila a un estado válido conocido anteriormente.
- **DELETE:** hace retroceder la pila a un estado válido conocido anteriormente, si es que existe alguno. De lo contrario, elimina la pila.
- **DO_NOTHING:** no revierte ni elimina la pila. El efecto es el mismo que el de `--disable-rollback`.

El comportamiento predeterminado es **ROLLBACK**.

Note

Especifica la opción `--disable-rollback` o la `--on-failure`, pero no ambas.

`--output` [*text* | *json*]

El formato de la salida del comando:

- **text**— Imprime un resultado normal y legible por humanos. Este es el valor predeterminado.
- **json**— Imprime resultados estructurados y legibles por máquina. Utilice este formato cuando desee consumir el resultado mediante programación, por ejemplo, en CI/CD canalizaciones, extensiones de IDE y herramientas. AI-assisted

Una implementación se ejecuta durante un período de tiempo. Cuando lo especificas `--output json`, AWS SAMCLI imprime una secuencia de objetos JSON, uno por línea, a medida que avanza la implementación. Cada objeto incluye un `type` campo que identifica el tipo de información que contiene, como el conjunto de cambios, los eventos de implementación, los resultados de la pila y el resultado final. En caso de error, el objeto final incluye un `error` objeto que describe lo que salió mal.

 Note

La `--output json` opción no es compatible con las opciones que requieren indicaciones interactivas, como `--guided` y `--confirm-changeset`.

`--parameter-overrides` *LIST*

Cadena que contiene anulaciones de CloudFormation parámetros codificadas como pares clave-valor. Cada anulación usa el formato `ParameterKey=name,ParameterValue=value`. Las anulaciones múltiples están separadas por espacios. A continuación, se incluyen dos ejemplos:

```
$ sam deploy --parameter-overrides ParameterKey=value1,ParameterValue=value2
```

```
$ sam deploy --parameter-overrides ParameterKey=value1,ParameterValue=value2  
ParameterKey=hello,ParameterValue=world ParameterKey=apple,ParameterValue=banana
```

`--profile` *TEXT*

El perfil específico de su archivo de credenciales que obtiene las credenciales. AWS

`--region` *TEXT*

El lugar en el Región de AWS que realizar el despliegue. Por ejemplo, `us-east-1`.

`--resolve-image-repos`

Crea automáticamente repositorios de Amazon ECR para usarlos en el empaquetado y la implementación de implementaciones no guiadas. Esta opción solo se aplica a funciones y capas con la especificación `PackageType: Image`. Si especificas la opción `--guided`, entonces la CLI de AWS SAM ignora `--resolve-image-repos`.

 Note

Si crea AWS SAM automáticamente algún repositorio de Amazon ECR para funciones o capas con esta opción y, posteriormente, borra esas funciones o capas de la AWS SAM plantilla, los repositorios de Amazon ECR correspondientes se eliminarán automáticamente.

--resolve-s3

Crea automáticamente un bucket de Amazon S3 para utilizarlo en el empaquetado y la implementación de implementaciones no guiadas. Si especificas la opción --guided, entonces la CLI AWS SAM ignora --resolve-s3. Si especificas las opciones --s3-bucket y --resolve-s3, entonces se produce un error.

--role-arn *TEXT*

El nombre de recurso de Amazon (ARN) de una función de IAM que CloudFormation se asume al aplicar el conjunto de cambios.

--s3-bucket *TEXT*

El nombre del bucket de Amazon S3 en el que este comando carga la plantilla. CloudFormation Si la plantilla tiene más de 51 200 bytes, es necesaria la opción --s3-bucket o la opción --resolve-s3. Si especificas ambas opciones --s3-bucket y --resolve-s3, se producirá un error.

--s3-prefix *TEXT*

El prefijo que se ha añadido a los nombres de los artefactos que se cargan en el bucket de Amazon S3. El nombre del prefijo es un nombre de ruta (nombre de la carpeta) del bucket de Amazon S3.

--save-params

Guarde los parámetros que proporciona en la línea de comandos en el archivo de AWS SAM configuración.

--signing-profiles *LIST*

La lista de perfiles de firma con los que firmar los paquetes de implementación. Esta opción utiliza una lista de pares clave-valor, donde la clave es el nombre de la función o capa que se va a firmar y el valor es el perfil de firma, con un propietario del perfil opcional delimitado con :. Por ejemplo, FunctionNameToSign=SigningProfileName1 LayerNameToSign=SigningProfileName2:SigningProfileOwner.

--stack-name *TEXT*

(Obligatorio) El nombre de la CloudFormation pila en la que vas a realizar la implementación. Si especificas una pila existente, el comando la actualiza. Si especificas una nueva pila, el comando la crea.

`--tags` *LIST*

Una lista de etiquetas para asociarlas a la pila que se crea o actualiza. CloudFormation también propaga estas etiquetas a los recursos de la pila que las admiten.

`--template-file`, `--template`, `-t` *PATH*

La ruta y el nombre del archivo donde se encuentra la AWS SAM plantilla.

 Note

Si especifica esta opción, solo AWS SAM implementa la plantilla y los recursos locales a los que apunta.

`--use-json`

Muestra el JSON de salida para la CloudFormation plantilla. El resultado predeterminado es YAML.

Ejemplo

Para ver un ejemplo y un tutorial detallados sobre el uso del subcomando `sam deploy`, consulte [Introducción a la implementación con AWS SAM](#).

sam init

Esta página proporciona información de referencia para el AWS Serverless Application Model comando Command Line Interface (AWS SAMCLI) `sam init`.

- Para obtener una introducción al AWS SAMCLI, consulte [¿Qué es el AWS SAM CLI?](#)
- Para obtener documentación sobre el uso del AWS SAMCLI `sam init` comando, consulte [Crea tu aplicación en AWS SAM](#).

El comando `sam init` proporciona opciones para inicializar una nueva aplicación sin servidor.

De uso

```
$ sam init <options>
```


Opciones

`--app-template` *TEXT*

El identificador de la plantilla de la aplicación gestionada que recomendamos utilizar. Si no está seguro, llame a `sam init` sin opciones para disfrutar de un flujo de trabajo interactivo.

Este parámetro es obligatorio si se especifica `--no-interactive`, y `--location` si no se proporciona.

Esta característica solo está disponible en la versión 0.30.0 o posterior de la CLI de AWS SAM. Si se especifica este parámetro con una versión anterior, se produce un error.

`--application-insights` | `--no-application-insights`

Active la supervisión de Amazon CloudWatch Application Insights para su aplicación. Para obtener más información, consulte [Uso de CloudWatch Application Insights para monitorear sus aplicaciones AWS SAM sin servidor](#).

La opción predeterminada es `--no-application-insights`.

`--architecture`, `-a` [*x86_64* | *arm64*]

La arquitectura del conjunto de instrucciones para las funciones de eventos Lambda de su aplicación. Especifica una de las siguientes opciones: `x86_64` o `arm64`

`--base-image` [*amazon/dotnet8-base* | *amazon/dotnet6-base* | *amazon/java25-base* | *amazon/java21-base* | *amazon/java17-base* | *amazon/java17.al2023-base* | *amazon/java11-base* | *amazon/java11.al2023-base* | *amazon/java8.al2023-base* | *amazon/nodejs24.x-base* | *amazon/nodejs22.x-base* | *amazon/nodejs20.x-base* | *amazon/nodejs18.x-base* | *amazon/nodejs16.x-base* | *amazon/python3.14-base* | *amazon/python3.13-base* | *amazon/python3.12-base* | *amazon/python3.11-base* | *amazon/python3.10-base* | *amazon/python3.9-base* | *amazon/python3.8-base* | *amazon/ruby4.0-base* | *amazon/ruby3.4-base* | *amazon/ruby3.3-base* | *amazon/ruby3.2-base*]

La imagen base de su aplicación. Esta opción solo se aplica cuando el tipo de paquete es `Image`.

Este parámetro es obligatorio si `--no-interactive` se especifica, `--package-type` se especifica como `Image` y `--location` no se especifica.

`--config-env` *TEXT*

El nombre del entorno que especifica los valores de los parámetros predeterminados del archivo de configuración que se va a utilizar. El valor por defecto es «predeterminado». Para obtener más información sobre los archivos de configuración, consulta [Archivo de configuración de la CLI de AWS SAM](#).

`--config-file` *PATH*

La ruta y el nombre del archivo de configuración que contiene los valores de parámetros predeterminados que se van a utilizar. El valor predeterminado es «samconfig.toml» en la raíz del directorio del proyecto. Para obtener más información sobre los archivos de configuración, consulta [Archivo de configuración de la CLI de AWS SAM](#).

`--debug`

Activa el registro de depuración para imprimir los mensajes de depuración que genera la AWS SAM de CLI y para mostrar las marcas temporales.

`--dependency-manager, -d` [*gradle* | *mod* | *maven* | *bundler* | *npm* | *cli-package* | *pip*]

El administrador de dependencias de su tiempo de ejecución de Lambda.

`--extra-content`

Anule cualquier parámetro personalizado de la configuración `cookiecutter.json` de la plantilla, por ejemplo, `{"customParam1": "customValue1", "customParam2": "customValue2"}`.

`--help, -h`

Muestra este mensaje y sale.

`--location, -l` *TEXT*

La ubicación de la plantilla o aplicación (Git, Mercurial HTTP/HTTPS, archivo.zip, ruta).

Este parámetro es obligatorio si se especifica `--no-interactive`, `--runtime` o `--name`, y `--app-template` si no se proporciona.

Para los repositorios Git, debe usar la ubicación de la raíz del repositorio.

Para las rutas locales, la plantilla debe estar en formato de archivo.zip o [Cookiecutter](#).

`--name, -n TEXT`

El nombre del proyecto que se generará como un directorio.

Este parámetro es obligatorio si se especifica `--no-interactive`, y `--location` si no se proporciona.

`--no-input`

Desactiva las solicitudes de Cookiecutter y acepta los valores `vcfdefault` que se definen en la configuración de plantilla.

`--no-interactive`

Deshabilita la solicitud interactiva para los parámetros de inicio y falla si falta algún valor obligatorio.

`--output [ text | json ]`

Formato de la salida del comando:

- `text`— Imprime un resultado normal y legible para los humanos. Este es el valor predeterminado.
- `json`— Imprime resultados estructurados y legibles por máquina. Utilice este formato cuando desee consumir la salida mediante programación, por ejemplo, en CI/CD canalizaciones, extensiones IDE y herramientas. `AI-assisted`

Si lo especificas `--output json`, AWS SAMCLI imprime un único objeto JSON una vez finalizado el comando. En caso de éxito, este objeto incluye campos como el directorio del proyecto y el archivo de plantilla. Puede encadenar la salida en comandos posteriores, como `sam build`. En caso de error, incluye un `error` objeto que describe lo que ha fallado.

 Note

La `--output json` opción no es compatible con el flujo de trabajo interactivo. Para usarla, especifique una ejecución totalmente no interactiva proporcionando todas las opciones necesarias.

`--output-dir, -o PATH`

La ubicación en la que se muestra la aplicación inicializada.

`--package-type [ Zip | Image ]`

El tipo de paquete de la aplicación de ejemplo. Zip crea un archivo .zip y Image crea una imagen del contenedor.

`--runtime, -r [dotnet8 | dotnet6 | java25 | java21 | java17 | java17.al2023 | java11 | java11.al2023 | java8.al2023 | nodejs24.x | nodejs22.x | nodejs20.x | nodejs18.x | nodejs16.x | python3.14 | python3.13 | python3.12 | python3.11 | python3.10 | python3.9 | python3.8 | ruby4.0 | ruby3.4 | ruby3.3 | ruby3.2]`

El tiempo de ejecución Lambda de su aplicación. Esta opción solo se aplica cuando el tipo de paquete es Zip.

Este parámetro es obligatorio si `--no-interactive` se especifica, `--package-type` se especifica como Zip y `--location` no se especifica.

`--save-params`

Guarde los parámetros que proporcione en la línea de comandos en el archivo de AWS SAM configuración.

`--tracing | --no-tracing`

Active el AWS X-Ray rastreo para sus funciones Lambda.

Ejemplo

Para ver un ejemplo y un tutorial detallados sobre el uso del subcomando `sam init`, consulte [Crea tu aplicación en AWS SAM](#).

sam list

Esta página proporciona información de referencia para el AWS Serverless Application Model comando Command Line Interface (AWS SAMCLI) `sam list`.

Para obtener una introducción al AWS SAMCLI, consulte [¿Qué es el AWS SAM CLI?](#)

El comando `sam list` genera información importante sobre los recursos de la aplicación sin servidor y el estado de la aplicación sin servidor. Usa `sam list` antes y después de la implementación para ayudar durante el desarrollo local y en la nube.

De uso

```
$ sam list <options> <subcommand>
```

Opciones

--help, -h

Muestre este mensaje y salga de él.

Subcomandos

endpoints

Muestra una lista de puntos finales locales y en la nube de su CloudFormation pila. Para obtener más información, consulte [sam list endpoints](#).

resources

Muestra los recursos de tu plantilla AWS Serverless Application Model (AWS SAM) que se crean en el AWS CloudFormation momento de la implementación. Para obtener más información, consulte [sam list resources](#).

stack-outputs

Muestra los resultados de la CloudFormation pila a partir de una CloudFormation plantilla AWS SAM o. Para obtener más información, consulte [sam list stack-outputs](#).

sam list endpoints

Esta página proporciona información de referencia para el AWS Serverless Application Model `sam list endpoints` subcomando Command Line Interface (AWS SAMCLI).

Para obtener una introducción al AWS SAMCLI, consulte [¿Qué es el AWS SAM CLI?](#)

El `sam list endpoints` subcomando muestra una lista de puntos finales locales y en la nube de su CloudFormation pila. Puede interactuar con estos recursos mediante los comandos `sam local` y `sam sync`.

AWS Lambda y los tipos de recursos de Amazon API Gateway son compatibles con este comando.

 Note

Los dominios personalizados se admiten cuando se configuran para sus recursos de Amazon API Gateway. Este comando generará el dominio personalizado en lugar del punto de conexión predeterminado.

De uso

```
$ sam list endpoints <options>
```

Opciones

`--config-env` *TEXT*

El nombre del entorno que especifica los valores de los parámetros predeterminados del archivo de configuración que se va a utilizar.

Valor predeterminado: `default`

Para obtener más información sobre los archivos de configuración, consulta [Archivo de configuración de la CLI de AWS SAM](#).

`--config-file` *TEXT*

La ruta y el nombre del archivo de configuración que contiene los valores de parámetros predeterminados que se van a utilizar.

Valor predeterminado: `samconfig.toml` en el directorio de trabajo actual.

Para obtener más información sobre los archivos de configuración, consulta [Archivo de configuración de la CLI de AWS SAM](#).

`--debug`

Active el registro de depuración para imprimir los mensajes de depuración generados por el AWS SAMCLI con marcas temporales.

`--help`, `-h`

Muestre este mensaje y salga de él.

--output [json|table]

Especifica el formato de salida de los resultados.

Valor predeterminado: table

--profile *TEXT*

Seleccione un perfil específico de su archivo de credenciales para obtener AWS las credenciales.

--region *TEXT*

Defina la AWS región del servicio. Por ejemplo, us-east-1.

--save-params

Guarde los parámetros que proporcione en la línea de comandos en el archivo AWS SAM de configuración.

--stack-name *TEXT*

Nombre de la CloudFormation pila implementada. El nombre de la pila se encuentra en el archivo de la aplicación o en el archivo de configuración samconfig.toml designado.

Si no se especifica esta opción, se mostrarán los recursos locales definidos en la plantilla.

--template-file, --template, -t *PATH*

AWS SAM archivo de plantilla.

Valor predeterminado: template.[yaml|yml|json]

Ejemplos

Muestre una salida, en formato json, de los puntos finales de recursos implementados de su CloudFormation pila con el nombre test-stack.

```
$ sam list endpoints --stack-name test-stack --output json

[
  {
    "LogicalResourceId": "HelloWorldFunction",
    "PhysicalResourceId": "sam-app-test-list-HelloWorldFunction-H85Y7yIV7ZLq",
    "CloudEndpoint": "https://zt55oi7kbljxjmcoahsj3cknwu0rposq.lambda-url.us-east-1.on.aws/",
```

```

    "Methods": "- "
  },
  {
    "LogicalResourceId": "ServerlessRestApi",
    "PhysicalResourceId": "uj80ue2o2",
    "CloudEndpoint": [
      "https://uj80ue2o2.execute-api.us-east-1.amazonaws.com/Prod",
      "https://uj80ue2o2.execute-api.us-east-1.amazonaws.com/Stage"
    ],
    "Methods": [
      "/hello['get']"
    ]
  }
]

```

sam list resources

Esta página proporciona información de referencia para el AWS Serverless Application Model `sam list resources` subcomando Command Line Interface (AWS SAMCLI).

Para obtener una introducción al AWS SAMCLI, consulte [¿Qué es el AWS SAM CLI?](#)

El `sam list resources` subcomando muestra los recursos de la plantilla AWS Serverless Application Model (AWS SAM) que se crean AWS CloudFormation mediante la AWS SAM transformación en el momento del despliegue.

`sam list resources` Utilícelo con una AWS SAM plantilla antes del despliegue para ver los recursos que se crearán. Proporcione un nombre de CloudFormation pila para ver una lista consolidada que incluya los recursos desplegados.

Note

Para generar una lista de recursos a partir de la AWS SAM plantilla, se realiza una transformación local de la plantilla. En esta lista se incluyen los recursos que se implementarán con determinadas condiciones, por ejemplo, en una región específica.

De uso

```
$ sam list resources <options>
```


Opciones

`--config-env` *TEXT*

El nombre del entorno que especifica los valores de los parámetros predeterminados del archivo de configuración que se va a utilizar.

Valor predeterminado: `default`

Para obtener más información sobre los archivos de configuración, consulta [Archivo de configuración de la CLI de AWS SAM](#).

`--config-file` *TEXT*

La ruta y el nombre del archivo de configuración que contiene los valores de parámetros predeterminados que se van a utilizar.

Valor predeterminado: `samconfig.toml` en el directorio de trabajo actual.

Para obtener más información sobre los archivos de configuración, consulta [Archivo de configuración de la CLI de AWS SAM](#).

`--debug`

Active el registro de depuración para imprimir los mensajes de depuración generados por el AWS SAMCLI con marcas temporales.

`--help`, `-h`

Muestre este mensaje y salga de él.

`--output` [`json`|`table`]

Especifica el formato de salida de los resultados.

Valor predeterminado: `table`

`--profile` *TEXT*

Seleccione un perfil específico de su archivo de credenciales para obtener AWS las credenciales.

`--region` *TEXT*

Defina la AWS región del servicio. Por ejemplo, `us-east-1`.

--save-params

Guarde los parámetros que proporcione en la línea de comandos en el archivo AWS SAM de configuración.

--stack-name *TEXT*

Nombre de la CloudFormation pila implementada. El nombre de la pila se encuentra en el archivo de la aplicación o en el archivo de configuración `samconfig.toml` designado.

Cuando se proporcione, la lógica IDs de recursos de la plantilla se asignará a su entrada física IDs correspondiente. CloudFormation Para obtener más información sobre la física IDs, consulte [los campos de recursos](#) en la Guía del AWS CloudFormation usuario.

Si no se especifica esta opción, se mostrarán los recursos locales definidos en la plantilla.

--template-file, --template, -t *PATH*

AWS SAM archivo de plantilla.

Valor predeterminado: `template.[yaml|yml|json]`

Ejemplos

Muestra una salida, en formato de tabla, de los recursos locales de tu AWS SAM plantilla y de los recursos desplegados de tu CloudFormation pila denominada `test-stack`. Se ejecuta en el mismo directorio que la plantilla local.

```
$ sam list resources --stack-name test-stack --output table
```

Logical ID	Physical ID
HelloWorldFunction	sam-app-test-list-
HelloWorldFunction-H85Y7yIV7ZLq	
HelloWorldFunctionHelloWorldPermissionProd	sam-app-test-list-
HelloWorldFunctionHelloWorldPermissionProd-1QH7CP0CBL2IK	
HelloWorldFunctionRole	sam-app-test-list-
HelloWorldFunctionRole-SRJDMJ6F7F41	
ServerlessRestApi	uj80uoe2o2
ServerlessRestApiDeployment47fc2d5f9d	pncw5f
ServerlessRestApiProdStage	Prod
ServerlessRestApiDeploymentf5716dc08b	-

sam list stack-outputs

Esta página proporciona información de referencia para el AWS Serverless Application Model `sam list stack-outputs` subcomando Command Line Interface (AWS SAMCLI).

Para obtener una introducción al AWS SAMCLI, consulte [¿Qué es el AWS SAM CLI?](#)

El `sam list stack-outputs` subcomando muestra los resultados de la AWS CloudFormation pila a partir de un AWS Serverless Application Model (AWS SAM) o una CloudFormation plantilla. Para obtener más información sobre Outputs, consulta [Salidas](#) en la Guía del usuario de AWS CloudFormation .

De uso

```
$ sam list stack-outputs <options>
```

Opciones

`--config-env` *TEXT*

El nombre del entorno que especifica los valores de los parámetros predeterminados del archivo de configuración que se va a utilizar.

Valor predeterminado: `default`

Para obtener más información sobre los archivos de configuración, consulta [Archivo de configuración de la CLI de AWS SAM](#).

`--config-file` *TEXT*

La ruta y el nombre del archivo de configuración que contiene los valores de parámetros predeterminados que se van a utilizar.

Valor predeterminado: `samconfig.toml` en el directorio de trabajo actual.

Para obtener más información sobre los archivos de configuración, consulta [Archivo de configuración de la CLI de AWS SAM](#).

`--debug`

Active el registro de depuración para imprimir los mensajes de depuración generados por el AWS SAMCLI con marcas temporales.

`--help, -h`

Muestre este mensaje y salga de él.

`--output [json|table]`

Especifica el formato de salida de los resultados.

Valor predeterminado: `table`

`--profile TEXT`

Seleccione un perfil específico de su archivo de credenciales para obtener AWS las credenciales.

`--region TEXT`

Defina la AWS región del servicio. Por ejemplo, `us-east-1`.

`--save-params`

Guarde los parámetros que proporcione en la línea de comandos en el archivo AWS SAM de configuración.

`--stack-name TEXT`

Nombre de la CloudFormation pila implementada. El nombre de la pila se encuentra en el archivo de la aplicación o en el archivo de configuración `samconfig.toml` designado.

Esta opción es obligatoria.

Ejemplos

Muestra los resultados, en formato de tabla, de los recursos nombrados en la CloudFormation `pilatest-stack`.

```
$ sam list stack-outputs --stack-name test-stack --output table
```

OutputKey	OutputValue
Description	
HelloWorldFunctionIamRole	arn:aws:iam:: <i>account-number</i> :role/sam-
Implicit IAM Role created for Hello	app-test-list-HelloWorldFunctionRole-
function	World
	SRJDMJ6F7F41

HelloWorldApi	https://uj80uoe2o2.execute-api.us-	API
Gateway endpoint URL for Prod	east-1.amazonaws.com/Prod/hello/	stage
for Hello World function		
HelloWorldFunction	arn:aws:lambda:us-	Hello
World Lambda Function ARN	east-1: <i>account-number</i> :function:sam-app-test-list-HelloWorldFunction-H85Y7yIV7ZLq	

sam local callback

Esta página proporciona información de referencia para el AWS Serverless Application Model comando Command Line Interface (AWS SAMCLI) `sam local callback`.

Para obtener una introducción al AWS SAMCLI, consulte [¿Qué es el AWS SAM CLI?](#)

El `sam local callback` comando permite enviar llamadas a la ejecución local de funciones duraderas durante el desarrollo y las pruebas.

De uso

```
$ sam local callback <subcommand> <options>
```

Opciones

`--help`, `-h`

Muestre este mensaje y salga de él.

Subcomandos

succeed

Envíe una llamada de éxito a la ejecución de una función duradera. Para obtener más información, consulte [sam local callback succeed](#).

fail

Envíe una llamada de error a la ejecución de una función duradera. Para obtener más información, consulte [sam local callback fail](#).

heartbeat

Envíe una llamada rápida a la ejecución de una función duradera. Para obtener más información, consulte [sam local callback heartbeat](#).

sam local callback succeed

Envíe una llamada de éxito a la ejecución de una función duradera.

De uso

```
$ sam local callback succeed CALLBACK_ID [OPTIONS]
```

Argumentos necesarios

CALLBACK_ID

El identificador único de la devolución de llamada a la que se debe enviar una respuesta correcta.

Opciones

-r, --result TEXT

El resultado fue exitoso: carga útil como cadena.

--region TEXT

Defina la AWS región del servicio (por ejemplo,us-east-1).

--profile TEXT

Seleccione un perfil específico de su archivo de credenciales para obtener AWS las credenciales.

--config-env TEXT

Nombre del entorno que especifica los valores de los parámetros predeterminados en el archivo de configuración. Valor predeterminado: default

--config-file TEXT

Archivo de configuración que contiene los valores de los parámetros predeterminados. Valor predeterminado: samconfig.toml

--save-params

Guarda los parámetros proporcionados a través de la línea de comandos en el archivo de configuración.

--beta-features / --no-beta-features

Activar/desactivar las funciones beta.

--debug

Active el registro de depuración para imprimir los mensajes de depuración generados por la AWS SAM CLI y mostrar las marcas de tiempo.

-h, --help

Muestra este mensaje y sale.

Ejemplos

Enviar una llamada de éxito sin resultado:

```
$ sam local callback succeed my-callback-id
```

Envía una llamada de confirmación con el resultado:

```
$ sam local callback succeed my-callback-id --result 'Task completed successfully'
```

Envía una llamada de éxito con una opción corta:

```
$ sam local callback succeed my-callback-id -r 'Success result'
```

sam local callback fail

Envía una llamada de error a la ejecución de una función duradera.

De uso

```
$ sam local callback fail CALLBACK_ID [OPTIONS]
```

Argumentos necesarios

CALLBACK_ID

El identificador único de la devolución de llamada a la que se va a enviar una respuesta de error.

Opciones

--error-data TEXT

Datos de error adicionales.

--stack-trace TEXT

Apila las entradas de rastreo. Esta opción se puede especificar varias veces.

--error-type TEXT

Tipo de error.

--error-message TEXT

Mensaje de error detallado.

--region TEXT

Defina la AWS región del servicio (por ejemplo,us-east-1).

--profile TEXT

Seleccione un perfil específico de su archivo de credenciales para obtener AWS las credenciales.

--config-env TEXT

Nombre del entorno que especifica los valores de los parámetros predeterminados en el archivo de configuración. Valor predeterminado: default

--config-file TEXT

Archivo de configuración que contiene los valores de los parámetros predeterminados. Valor predeterminado: samconfig.toml

--save-params

Guarde los parámetros proporcionados a través de la línea de comandos en el archivo de configuración.

--beta-features / --no-beta-features

Activar/desactivar las funciones beta.

--debug

Active el registro de depuración para imprimir los mensajes de depuración generados por la AWS SAM CLI y mostrar las marcas de tiempo.

-h, --help

Muestra este mensaje y sale.

Ejemplos

Envía una llamada de error sin parámetros:

```
$ sam local callback fail my-callback-id
```

Envía una llamada de error con un mensaje de error:

```
$ sam local callback fail my-callback-id --error-message 'Task failed'
```

Envía una llamada de error con todos los parámetros:

```
$ sam local callback fail my-callback-id --error-message 'Task failed' --error-type 'ValidationError' --stack-trace 'at line 42' --error-data '{"code": 500}'
```

sam local callback heartbeat

Envíe una llamada de latido a la ejecución de una función duradera.

De uso

```
$ sam local callback heartbeat CALLBACK_ID [OPTIONS]
```

Argumentos necesarios

CALLBACK_ID

El identificador único de la devolución de llamada a la que se va a enviar un latido.

Opciones

`--region TEXT`

Defina la AWS región del servicio (por ejemplo, `us-east-1`).

`--profile TEXT`

Seleccione un perfil específico de su archivo de credenciales para obtener AWS las credenciales.

`--config-env TEXT`

Nombre del entorno que especifica los valores de los parámetros predeterminados en el archivo de configuración. Valor predeterminado: `default`

`--config-file TEXT`

Archivo de configuración que contiene los valores de los parámetros predeterminados. Valor predeterminado: `samconfig.toml`

`--save-params`

Guarde los parámetros proporcionados a través de la línea de comandos en el archivo de configuración.

`--beta-features / --no-beta-features`

Activar/desactivar las funciones beta.

`--debug`

Active el registro de depuración para imprimir los mensajes de depuración generados por la AWS SAM CLI y mostrar las marcas de tiempo.

`-h, --help`

Muestra este mensaje y sale.

Ejemplos

Enviar una llamada de latido:

```
$ sam local callback heartbeat my-callback-id
```

sam local execution

Esta página proporciona información de referencia para el AWS Serverless Application Model comando Command Line Interface (AWS SAMCLI) `sam local execution`.

Para obtener una introducción al AWS SAMCLI, consulte [¿Qué es el AWS SAM CLI?](#)

El `sam local execution` comando permite gestionar e inspeccionar las ejecuciones locales de funciones duraderas durante el desarrollo y las pruebas.

Note

Es posible que estos comandos no requieran acceso a AWS las credenciales.

De uso

```
$ sam local execution <subcommand> <options>
```

Opciones

`--help`, `-h`

Muestre este mensaje y salga de él.

Subcomandos

get

Obtenga detalles de la ejecución de una función duradera. Para obtener más información, consulte [sam local execution get](#).

history

Obtenga el historial de ejecución de una función duradera. Para obtener más información, consulte [sam local execution history](#).

stop

Detenga la ejecución duradera de una función. Para obtener más información, consulte [sam local execution stop](#).

sam local execution get

Recupera los detalles de la ejecución de una función duradera específica.

Note

Es posible que este comando no requiera acceso a AWS las credenciales.

De uso

```
$ sam local execution get DURABLE_EXECUTION_ARN [OPTIONS]
```

Argumentos necesarios

DURABLE_EXECUTION_ARN

El nombre del recurso de Amazon (ARN) de la ejecución de la función duradera de la que se van a recuperar los detalles.

Opciones

--format [summary|json]

Formato de salida. Valor predeterminado: summary

--region TEXT

Defina la AWS región del servicio (por ejemplo,us-east-1).

--profile TEXT

Seleccione un perfil específico de su archivo de credenciales para obtener AWS las credenciales.

--config-env TEXT

Nombre del entorno que especifica los valores de los parámetros predeterminados en el archivo de configuración. Valor predeterminado: default

--config-file TEXT

Archivo de configuración que contiene los valores de los parámetros predeterminados. Valor predeterminado: samconfig.toml

--save-params

Guarde los parámetros proporcionados a través de la línea de comandos en el archivo de configuración.

--beta-features / --no-beta-features

Activar/desactivar las funciones beta.

--debug

Active el registro de depuración para imprimir los mensajes de depuración generados por la AWS SAM CLI y mostrar las marcas de tiempo.

-h, --help

Muestra este mensaje y sale.

Ejemplos

Obtenga los detalles de la ejecución con el formato de resumen predeterminado:

```
$ sam local execution get arn:aws:lambda:us-east-1:123456789012:function:MyFunction:
$LATEST/durable-execution/my-execution-name/my-execution-id
```

Obtenga los detalles de la ejecución en formato JSON:

```
$ sam local execution get arn:aws:lambda:us-east-1:123456789012:function:MyFunction:
$LATEST/durable-execution/my-execution-name/my-execution-id --format json
```

sam local execution history

Obtenga el historial de ejecución de una función duradera.

De uso

```
$ sam local execution history EXECUTION_ID [OPTIONS]
```

Argumentos necesarios

EXECUTION_ID

El identificador único de la ejecución duradera de la función de la que se va a recuperar el historial.

Opciones

`--format [table|json]`

Formato de salida. Valor predeterminado: `table`

`--region TEXT`

Defina la AWS región del servicio (por ejemplo, `us-east-1`).

`--profile TEXT`

Seleccione un perfil específico de su archivo de credenciales para obtener AWS las credenciales.

`--config-env TEXT`

Nombre del entorno que especifica los valores de los parámetros predeterminados en el archivo de configuración. Valor predeterminado: `default`

`--config-file TEXT`

Archivo de configuración que contiene los valores de los parámetros predeterminados. Valor predeterminado: `samconfig.toml`

`--save-params`

Guarde los parámetros proporcionados a través de la línea de comandos en el archivo de configuración.

`--beta-features / --no-beta-features`

Activar/desactivar las funciones beta.

`--debug`

Active el registro de depuración para imprimir los mensajes de depuración generados por la AWS SAM CLI y mostrar las marcas de tiempo.

`-h, --help`

Muestra este mensaje y sale.

Ejemplos

Obtenga el historial de ejecuciones en formato de tabla:

```
$ sam local execution history my-execution-id
```

Obtenga el historial de ejecuciones en formato JSON:

```
$ sam local execution history my-execution-id --format json
```

`sam local execution stop`

Detenga la ejecución de una función duradera en ejecución.

 Note

Es posible que este comando no requiera acceso a AWS las credenciales.

De uso

```
$ sam local execution stop DURABLE_EXECUTION_ARN [OPTIONS]
```

Argumentos necesarios

DURABLE_EXECUTION_ARN

El nombre del recurso de Amazon (ARN) de la ejecución duradera de la función que se va a detener.

Opciones

--error-message TEXT

Mensaje de error asociado a la ejecución detenida.

--error-type TEXT

Tipo de error que se va a asociar a la ejecución detenida.

--error-data TEXT

Datos de error que se van a asociar a la ejecución detenida.

--stack-trace TEXT

Apila las entradas de rastreo. Esta opción se puede especificar varias veces.

--region TEXT

Defina la AWS región del servicio (por ejemplo, `us-east-1`).

--profile TEXT

Seleccione un perfil específico de su archivo de credenciales para obtener AWS las credenciales.

--config-env TEXT

Nombre del entorno que especifica los valores de los parámetros predeterminados en el archivo de configuración. Valor predeterminado: `default`

--config-file TEXT

Archivo de configuración que contiene los valores de los parámetros predeterminados. Valor predeterminado: `samconfig.toml`

--save-params

Guarde los parámetros proporcionados a través de la línea de comandos en el archivo de configuración.

--beta-features / --no-beta-features

Activar/desactivar las funciones beta.

--debug

Active el registro de depuración para imprimir los mensajes de depuración generados por la AWS SAM CLI y mostrar las marcas de tiempo.

-h, --help

Muestra este mensaje y sale.

Ejemplos

Detenga la ejecución sin detalles del error:

```
$ sam local execution stop arn:aws:lambda:us-east-1:123456789012:function:MyFunction:
$LATEST/durable-execution/my-execution-name/my-execution-id
```

Detenga la ejecución con un mensaje de error y escriba:


```
$ sam local execution stop arn:aws:lambda:us-east-1:123456789012:function:MyFunction:
$LATEST/durable-execution/my-execution-name/my-execution-id --error-message "Execution
cancelled" --error-type "UserCancellation"
```

Detenga la ejecución con todos los detalles del error y el seguimiento de la pila:

```
$ sam local execution stop arn:aws:lambda:us-east-1:123456789012:function:MyFunction:
$LATEST/durable-execution/my-execution-name/my-execution-id --error-message "Task
failed" --error-type "TaskFailure" --error-data '{"reason":"timeout"}' --stack-trace
"at function1()" --stack-trace "at function2()"
```

sam local generate-event

Esta página proporciona información de referencia para el AWS Serverless Application Model `sam local generate-event` subcomando de la interfaz de línea de comandos (AWS SAMCLI).

- Para obtener una introducción al AWS SAMCLI, consulte [¿Qué es el AWS SAM CLI?](#)
- Para obtener documentación sobre el uso del AWS SAMCLI `sam local generate-event` comando, consulte [Introducción a las pruebas con generate-event en local en sam](#).

El subcomando `sam local generate-event` genera muestras de carga útil de eventos para los Servicios de AWS compatibles.

De uso

```
$ sam local generate-event <options> <service> <event> <event-options>
```

Opciones

`--config-env` TEXT

El nombre del entorno que especifica los valores de los parámetros predeterminados del archivo de configuración que se va a utilizar. El valor por defecto es «predeterminado». Para obtener más información sobre los archivos de configuración, consulta [Archivo de configuración de la CLI de AWS SAM](#).

`--config-file` PATH

La ruta y el nombre del archivo de configuración que contiene los valores de parámetros predeterminados que se van a utilizar. El valor por defecto es `samconfig.toml` en la raíz

del directorio del proyecto. Para obtener más información sobre los archivos de configuración, consulta [Archivo de configuración de la CLI de AWS SAM](#).

--help

Muestra este mensaje y sale.

Servicio

Para ver una lista de los servicios compatibles, ejecuta lo siguiente:

```
$ sam local generate-event
```

Event

Para ver una lista de los eventos compatibles que se pueden generar para cada servicio, ejecuta lo siguiente:

```
$ sam local generate-event <service>
```

Opciones de eventos

Para ver una lista de las opciones de eventos compatibles que puede modificar, ejecuta lo siguiente:

```
$ sam local generate-event <service> <event> --help
```

Ejemplos

Para ver ejemplos sobre el uso del subcomando `sam local generate-event`, consulte [Generar eventos de ejemplo](#).

invocar sam local

Esta página proporciona información de referencia para el AWS Serverless Application Model `sam local invoke` subcomando de la interfaz de línea de comandos (AWS SAMCLI).

- Para obtener una introducción al AWS SAMCLI, consulte [¿Qué es el AWS SAM CLI?](#)
- Para obtener documentación sobre el uso del AWS SAMCLI `sam local invoke` subcomando, consulte [Introducción a las pruebas con invocar sam local](#).

El `sam local invoke` subcomando inicia una única invocación local de una función de evento AWS Lambda .

 Note

No se recomienda utilizar las capacidades de invocación local de la CLI de SAM en código que no sea de confianza. Para tener un aislamiento completo del entorno local, ejecute el código directamente en el servicio Lambda.

 Note

Para que las funciones sean duraderas, `sam local invoke` admite la ejecución automática con puntos de control y reproducción automáticos. El contenedor permanece en funcionamiento durante la ejecución duradera de la función para gestionar la persistencia y la reanudación del estado.

De uso

```
$ sam local invoke <arguments> <options>
```

 Note

Si tiene más de una función definida en la AWS SAM plantilla, proporcione el identificador lógico de la función que desee invocar.

Argumentos

Resource ID (ID de recurso)

El ID de la función de Lambda que se invocará.

Este argumento es opcional. Si la aplicación contiene una sola función Lambda, la AWS SAM CLI la invocará. Si la aplicación contiene varias funciones, proporcione el ID de la función que se va a invocar.

Valores válidos: el identificador lógico o el ARN del recurso.

Opciones

`--add-host` *LIST*

Pasa una asignación de nombre de host a dirección IP al archivo de host del contenedor de Docker. Este parámetro se puede pasar varias veces.

Example

Ejemplo: `--add-host` *example.com:127.0.0.1*

`--beta-features` | `--no-beta-features`

Permitir o denegar las características beta.

`--config-env` *TEXT*

El nombre del entorno que especifica los valores de los parámetros predeterminados del archivo de configuración que se va a utilizar. El valor por defecto es «predeterminado». Para obtener más información sobre los archivos de configuración, consulta [Archivo de configuración de la CLI de AWS SAM](#).

`--config-file` *PATH*

La ruta y el nombre del archivo de configuración que contiene los valores de parámetros predeterminados que se van a utilizar. El valor por defecto es «samconfig.toml» en la raíz del directorio del proyecto. Para obtener más información sobre los archivos de configuración, consulta [Archivo de configuración de la CLI de AWS SAM](#).

`--container-env-vars`

(Opcional) Pase las variables de entorno al contenedor de imágenes de la función de Lambda al depurar localmente.

`--container-host` *TEXT*

Host del contenedor Lambda emulado localmente. El valor predeterminado es `localhost`. Si quiere ejecutar la CLI de AWS SAM en un contenedor Docker en macOS, puede especificar `host.docker.internal`. Si desea ejecutar el contenedor en un host diferente al de AWS SAMCLI, puede especificar la dirección IP del host remoto.

`--container-host-interface` *TEXT*

La dirección IP de la interfaz de red del host a la que deben enlazarse los puertos del contenedor. El valor predeterminado es 127.0.0.1. Utiliza 0.0.0.0 para enlazarse a todas las interfaces.

`--container-dns` *TEXT*

Especifica un servidor DNS personalizado para el contenedor de Docker. Este parámetro se puede pasar varias veces para especificar varios servidores DNS.

Example

Ejemplo: `--container-dns 8.8.8.8`

`--debug`

Activa el registro de depuración para imprimir los mensajes de depuración que genera la CLI de AWS SAM y para mostrar las marcas horarias.

`--debug-args` *TEXT*

Argumentos opcionales que se pasan al depurador.

`--debug-port, -d` *TEXT*

Cuando se especifica, inicia el contenedor de funciones de Lambda en modo de depuración y expone este puerto en el host local.

`--debugger-path` *TEXT*

La ruta del host a un depurador montado en el contenedor de Lambda.

`--docker-network` *TEXT*

El nombre o el ID de una red de Docker existente a la que deben conectarse los contenedores de Lambda y Docker, junto con la red en modo puente predeterminada. Si esto no se especifica, los contenedores de Lambda se conectan únicamente a la red Docker en modo puente predeterminada.

`--docker-volume-basedir, -v` *TEXT*

La ubicación del directorio base en el que se encuentra el AWS SAM archivo. Si Docker se ejecuta en una máquina remota, debe montar la ruta en la que se encuentra el AWS SAM archivo en la máquina Docker y modificar este valor para que coincida con la máquina remota.

`--durable-execution-name` *TEXT*

Nombre de la ejecución duradera (solo para funciones duraderas)

`--env-vars, -n` *PATH*

El JSON o el `.env` archivo que contiene los valores de las variables de entorno de la función Lambda. El formato del archivo se detecta automáticamente. Para obtener más información sobre los archivos de variable de entorno, consulta [Archivo de variables de entorno](#).

`--event, -e` *PATH*

El archivo JSON que contiene los datos de eventos que se pasan a la función de Lambda cuando se invoca. Si no especifica esta opción, no se asume ningún evento. Para introducir JSON desde `stdin`, debe introducir el valor «-». Para obtener más información sobre los formatos de los mensajes de eventos de AWS los distintos servicios, consulte [Trabajar con otros servicios](#) en la Guía para AWS Lambda desarrolladores.

`--force-image-build`

Especifica si la CLI de AWS SAM debe reconstruir la imagen utilizada para invocar funciones de Lambda con capas.

`--help`

Muestra este mensaje y sale.

`--hook-name` *TEXT*

El nombre del enlace que se utiliza para ampliar la funcionalidad de la CLI de AWS SAM.

Valores aceptados: `terraform`.

`--invoke-image` *TEXT*

El URI de la imagen del contenedor que quieres usar para la invocación de la función local.

Para las funciones Lambda con `PackageType: Zip`, esto anula la imagen base del tiempo de ejecución. De forma predeterminada, AWS SAM extrae la imagen del contenedor de Amazon ECR Public (que están listadas en [Repositorios de imágenes para AWS SAM](#)). Utiliza esta opción para extraer la imagen desde otra ubicación.

En el caso de las funciones Lambda con `PackageType: Image`, esto anula la plantilla `ImageUri` creada. Utilice esta opción para invocar localmente una imagen de aplicación diferente sin modificar la plantilla.

Puedes especificar esta opción varias veces. Cada instancia de esta opción puede tomar una cadena o un par clave-valor. Si especificas una cadena, será el URI de la imagen del contenedor que se usará en todas las funciones de la aplicación. Por ejemplo, `sam local invoke --invoke-image public.ecr.aws/sam/emu-python3.13`. Si especificas un par clave-valor, la clave es el nombre del recurso y el valor es el URI de la imagen del contenedor que se va a usar para ese recurso. Por ejemplo, `sam local invoke --invoke-image public.ecr.aws/sam/emu-python3.13 --invoke-image Function1=amazon/aws-sam-cli-emulation-image-python3.13`. Con los pares clave-valor, puede especificar distintas imágenes del contenedor para distintos recursos.

`--layer-cache-basedir` *DIRECTORY*

Especifica la ubicación del directorio base en el que se descargan las capas que utiliza la plantilla.

`--log-file, -l` *TEXT*

El archivo de registro al que se van a enviar los registros de tiempo de ejecución.

`--mount-symlinks`

Garantiza que la CLI de AWS SAM siempre monte los enlaces simbólicos que están presentes en los archivos para compilarlos o invocarlos. Esto solo se aplica a los enlaces simbólicos del directorio de nivel superior (es decir, los que se encuentran directamente en la raíz de la función). Por defecto, los enlaces simbólicos no se montan, excepto los necesarios para utilizar `build-in-source` para `node_modules` en NodeJS.

`--no-event`

Invoca la función con un evento vacío.

`--no-memory-limit`

Elimina la limitación de memoria en el contenedor durante la invocación local, incluso cuando la memoria está configurada en la AWS SAM plantilla.

`--parameter-overrides`

Cadena que contiene anulaciones de CloudFormation parámetros codificadas como pares clave-valor. Usa el mismo formato que (). AWS Command Line Interface AWS CLI El formato de la CLI de AWS SAM es de palabras clave y de valor explícitas, y cada modificación está separada por un espacio. A continuación, se incluyen dos ejemplos:

- `--parameter-overrides ParameterKey=hello,ParameterValue=world`

- `--parameter-overrides ParameterKey=hello,ParameterValue=world
ParameterKey=example1,ParameterValue=example2
ParameterKey=apple,ParameterValue=banana`

`--profile` *TEXT*

El perfil específico del archivo de credenciales que recibe AWS las credenciales.

`--region` *TEXT*

La AWS región en la que se va a realizar el despliegue. Por ejemplo, us-east-1.

`--runtime` *TEXT*

Utiliza el tiempo de ejecución especificado para invocar una función de Lambda de forma local. Esto anula el tiempo de ejecución definido en el archivo `template.yml`. Esto también permite probar funciones de Lambda con diferentes tiempos de ejecución sin modificar la configuración original de la función.

`--save-params`

Guarde los parámetros que proporcione en la línea de comandos en el archivo AWS SAM de configuración.

`--shutdown`

Emula un evento de apagado una vez finalizada la invocación, para probar la forma en que la extensión gestiona el comportamiento del apagado.

`--skip-prepare-infra`

Omite la fase de preparación si no se ha realizado ningún cambio en la infraestructura. Ejecuta con la opción `--hook-name`.

`--skip-pull-image`

De forma predeterminada, la CLI de AWS SAM comprueba el tiempo de ejecución remoto más reciente de Lambda y actualiza la imagen local automáticamente para mantenerla sincronizada.

Especifica esta opción para omitir la extracción de la imagen de Docker más reciente de su tiempo de ejecución de Lambda.

`--template, -t` *PATH*

El archivo AWS SAM de plantilla.

Esta opción no es compatible con `--hook-name`.

 Note

Si especifica esta opción, AWS SAM carga únicamente la plantilla y los recursos locales a los que apunta.

`--tenant-id` *TEXT*

El ID de inquilino de las funciones Lambda multiusuario. Se utiliza para garantizar el aislamiento informático entre distintos inquilinos. Se requiere cuando se invocan funciones configuradas con el modo de aislamiento de inquilinos.

`--terraform-plan-file`

La ruta relativa o absoluta al archivo del plan local Terraform cuando se utiliza la CLI de AWS SAM con Terraform Cloud. Esta opción requiere que `--hook-name` esté configurado en terraform.

Ejemplos

En el siguiente ejemplo, se utiliza un evento generado para realizar pruebas locales mediante un evento `s3.json` para invocar una función de Lambda de forma local.

```
$ sam local invoke --event events/s3.json S3JsonLoggerFunction
```

El siguiente ejemplo prueba la función `HelloWorldFunction` con el tiempo de ejecución de Python 3.11

```
$ sam local invoke --runtime python3.11 HelloWorldFunction
```

El siguiente ejemplo prueba la función `HelloWorldFunction` con un nombre de ejecución duradero

```
$ sam local invoke HelloWorldFunction --durable-execution-name my-execution
```

sam local start-api

Esta página proporciona información de referencia para el AWS Serverless Application Model `sam local start-api` subcomando de la interfaz de línea de comandos (AWS SAMCLI).

- Para obtener una introducción al AWS SAMCLI, consulte [¿Qué es el AWS SAM CLI?](#)
- Para obtener documentación sobre el uso del AWS SAMCLI `sam local start-api` subcomando, consulte [Introducción a las pruebas con sam local start-api](#).

El `sam local start-api` subcomando ejecuta las funciones de AWS Lambda eventos de forma local para probarlas a través de un servidor HTTP local.

De uso

```
$ sam local start-api <options>
```

Opciones

`--add-host` *LIST*

Pasa una asignación de nombre de host a dirección IP al archivo de host del contenedor de Docker. Este parámetro se puede pasar varias veces.

Example

Ejemplo: `--add-host` *example.com:127.0.0.1*

`--beta-features` | `--no-beta-features`

Permitir o denegar las características beta.

`--config-env` *TEXT*

El nombre del entorno que especifica los valores de los parámetros predeterminados del archivo de configuración que se va a utilizar. El valor por defecto es «predeterminado». Para obtener más información sobre los archivos de configuración, consulta [Archivo de configuración de la CLI de AWS SAM](#).

`--config-file` *PATH*

La ruta y el nombre del archivo de configuración que contiene los valores de parámetros predeterminados que se van a utilizar. El valor predeterminado es «samconfig.toml» en la raíz del directorio del proyecto. Para obtener más información sobre los archivos de configuración, consulta [Archivo de configuración de la CLI de AWS SAM](#).

--container-env-vars

Opcional. Al realizar la depuración local, pasa las variables de entorno al contenedor de imágenes.

--container-host *TEXT*

Host del contenedor Lambda emulado localmente. El valor predeterminado es `localhost`. Si quiere ejecutar la CLI de AWS SAM en un contenedor Docker en macOS, puede especificar `host.docker.internal`. Si desea ejecutar el contenedor en un host diferente al de AWS SAMCLI, puede especificar la dirección IP del host remoto.

--container-host-interface *TEXT*

La dirección IP de la interfaz de red del host a la que deben enlazarse los puertos del contenedor. El valor predeterminado es `127.0.0.1`. Utiliza `0.0.0.0` para enlazarse a todas las interfaces.

--container-dns *TEXT*

Especifica un servidor DNS personalizado para el contenedor de Docker. Este parámetro se puede pasar varias veces para especificar varios servidores DNS.

Example

Ejemplo: **--container-dns 8.8.8.8**

--debug

Activa el registro de depuración para imprimir el mensaje de depuración que genera la CLI de AWS SAM y muestra las marcas temporales.

--debug-args *TEXT*

Los argumentos adicionales que se pasarán al depurador.

--debug-function

Opcional. Especifica la función de Lambda a la que se aplicarán las opciones de depuración cuando `--warm-containers` se especifica. Este parámetro se aplica a `--debug-port`, `--debugger-path` y `--debug-args`.

--debug-port, -d *TEXT*

Cuando se especifica, inicia el contenedor de funciones de Lambda en modo de depuración y expone este puerto en el host local.

`--debugger-path` *TEXT*

La ruta del host a un depurador que se montará en el contenedor de Lambda.

`--docker-network` *TEXT*

Especifica el nombre o el ID de una red de Docker existente a la que deben conectarse los contenedores de Lambda y Docker, junto con la red en modo puente predeterminada. Si esto no se especifica, los contenedores de Lambda solo se conectan a la red Docker en modo puente predeterminada.

`--docker-volume-basedir, -v` *TEXT*

La ubicación del directorio base en el que se encuentra el AWS SAM archivo. Si Docker se ejecuta en una máquina remota, debe montar la ruta en la que se encuentra el AWS SAM archivo en la máquina Docker y modificar este valor para que coincida con la máquina remota.

`--env-vars, -n` *PATH*

El JSON o el .env archivo que contiene los valores de las variables de entorno de la función Lambda. El formato del archivo se detecta automáticamente.

`--force-image-build`

Especifica si se AWS SAM CLI debe reconstruir la imagen utilizada para invocar funciones con capas.

`--help`

Muestra este mensaje y sale.

`--hook-name` *TEXT*

El nombre del enlace que se utiliza para ampliar la funcionalidad de la CLI de AWS SAM.

Valores aceptados: terraform.

`--host` *TEXT*

El nombre de host local o la dirección IP a la que se va a vincular (predeterminado: «127.0.0.1»).

`--invoke-image` *TEXT*

El URI de la imagen del contenedor que quiere usar para las funciones de Lambda.

Para las funciones Lambda con `PackageType: Zip`, esto anula la imagen base del tiempo de ejecución. De forma predeterminada, AWS SAM extrae la imagen del contenedor de Amazon ECR Public. Utiliza esta opción para extraer la imagen desde otra ubicación.

En el caso de las funciones Lambda con `PackageType: Image`, esto anula la plantilla `ImageUri` creada. Utilice esta opción para invocar localmente una imagen de aplicación diferente sin modificar la plantilla.

Puedes especificar esta opción varias veces. Cada instancia de esta opción puede tomar una cadena o un par clave-valor. Si especificas una cadena, será el URI de la imagen del contenedor que se usará en todas las funciones de la aplicación. Por ejemplo, `sam local start-api --invoke-image public.ecr.aws/sam/emu-python3.13`. Si especificas un par clave-valor, la clave es el nombre del recurso y el valor es el URI de la imagen del contenedor que se va a usar para ese recurso. Por ejemplo, `sam local start-api --invoke-image public.ecr.aws/sam/emu-python3.13 --invoke-image Function1=amazon/aws-sam-cli-emulation-image-python3.13`. Con los pares clave-valor, puede especificar distintas imágenes de contenedor para distintos recursos.

`--layer-cache-basedir` *DIRECTORY*

Especifica la ubicación basada en el lugar en el que se descargan las capas que utiliza la plantilla.

`--log-file, -l` *TEXT*

El archivo de registro al que se van a enviar los registros de tiempo de ejecución.

`--no-memory-limit`

Elimina la limitación de memoria en el contenedor durante la invocación local, incluso cuando la memoria está configurada en la AWS SAM plantilla.

`--no-watch`

Desactiva la visualización de archivos cuando se usa con. `--warm-containers` Cuando se especifica, AWS SAMCLI no supervisa el código fuente o la plantilla para detectar cambios, y los contenedores calientes no se reinician automáticamente cuando se modifica el código. Para detectar los cambios locales, detenga el comando y vuelva a ejecutarlo.

Utilice esta opción cuando la supervisión de archivos provoque una I/O sobrecarga o sobrecarga de la CPU, como en monorepos de gran tamaño, entornos con análisis de software antivirus o proyectos en los que varias funciones de Lambda comparten un. `CodeUri`

Esta opción se ignora si no `--warm-containers` se especifica.

--parameter-overrides

Cadena que contiene anulaciones de CloudFormation parámetros codificadas como pares clave-valor. Usa el mismo formato que (). AWS Command Line Interface AWS CLI El formato de la CLI de AWS SAM es de palabras clave y de valor explícitas, y cada modificación está separada por un espacio. A continuación, se incluyen dos ejemplos:

- --parameter-overrides ParameterKey=hello,ParameterValue=world
- --parameter-overrides ParameterKey=hello,ParameterValue=world
ParameterKey=example1,ParameterValue=example2
ParameterKey=apple,ParameterValue=banana

--port, -p *INTEGER*

El número de puerto local en el que se va a escuchar (valor predeterminado: «3000»).

--profile *TEXT*

El perfil específico del archivo de credenciales que recibe AWS las credenciales.

--region *TEXT*

La AWS región en la que se va a realizar el despliegue. Por ejemplo, us-east-1.

--save-params

Guarde los parámetros que proporcione en la línea de comandos en el archivo AWS SAM de configuración.

--shutdown

Emula un evento de apagado una vez finalizada la invocación, para probar la forma en que la extensión gestiona el comportamiento del apagado.

--skip-prepare-infra

Omite la fase de preparación si no se ha realizado ningún cambio en la infraestructura. Úsalo con la opción --hook-name.

--skip-pull-image

Especifica si la CLI debe omitir la extracción de la última imagen de Docker para el tiempo de ejecución de Lambda.

--ssl-cert-file *PATH*

Ruta al archivo de certificado SSL (predeterminado: ninguno). Al usar esta opción, también debes usarse la opción --ssl-key-file.

`--ssl-key-file` *PATH*

Ruta al archivo de clave SSL (predeterminado: ninguno). Al usar esta opción, también debes usarse la opción `--ssl-cert-file`.

`--static-dir, -s` *TEXT*

Todos los archivos de activos estáticos (por ejemplo, CSS/JavaScript/HTML) ubicados en este directorio se presentan en/.

`--template, -t` *PATH*

El archivo AWS SAM de plantilla.

 Note

Si especifica esta opción, AWS SAM carga únicamente la plantilla y los recursos locales a los que apunta.

`--terraform-plan-file`

La ruta relativa o absoluta al archivo del plan local Terraform cuando se utiliza la CLI de AWS SAM con Terraform Cloud. Esta opción requiere que `--hook-name` esté configurado en `terraform`.

`--warm-containers` *[EAGER | LAZY]*

Opcional. Especifica cómo la CLI de AWS SAM administra los contenedores para cada función.

Hay dos opciones disponibles:

EAGER: los contenedores de todas las funciones se cargan al inicio y se conservan entre las invocaciones.

LAZY: los contenedores solo se cargan cuando se invoca cada función por primera vez. Esos contenedores se conservan para realizar más invocaciones.

Ejemplo

En el siguiente ejemplo se inicia un servidor local, lo que le permite probar la aplicación mediante la API. Para que este comando funcione, la aplicación debe estar instalada y Docker debe estar en ejecución.

```
$ sam local start-api --port 3000
```

sam local start-lambda

Esta página proporciona información de referencia para el AWS Serverless Application Model `sam local start-lambda` subcomando de la interfaz de línea de comandos (AWS SAMCLI).

- Para obtener una introducción al AWS SAMCLI, consulte [¿Qué es el AWS SAM CLI?](#)
- Para obtener documentación sobre el uso del AWS SAMCLI `sam local start-lambda` subcomando, consulte [Introducción a las pruebas con sam local start-lambda](#).

El subcomando `sam local start-lambda` inicia un punto de conexión local para emular AWS Lambda.

De uso

```
$ sam local start-lambda <options>
```

Opciones

`--add-host` *LIST*

Pasa una asignación de nombre de host a dirección IP al archivo de host del contenedor de Docker. Este parámetro se puede pasar varias veces.

Example

Ejemplo: `--add-host` *example.com:127.0.0.1*

`--beta-features` | `--no-beta-features`

Permitir o denegar las características beta.

`--config-env` *TEXT*

El nombre del entorno que especifica los valores de los parámetros predeterminados del archivo de configuración que se va a utilizar. El valor por defecto es «predeterminado». Para obtener más información sobre los archivos de configuración, consulta [Archivo de configuración de la CLI de AWS SAM](#).

`--config-file` *PATH*

La ruta y el nombre del archivo de configuración que contiene los valores de parámetros predeterminados que se van a utilizar. El valor predeterminado es «samconfig.toml» en la raíz del directorio del proyecto. Para obtener más información sobre los archivos de configuración, consulta [Archivo de configuración de la CLI de AWS SAM](#).

`--container-env-vars`

Opcional. Al realizar la depuración local, pasa las variables de entorno al contenedor de imágenes.

`--container-host` *TEXT*

Host del contenedor Lambda emulado localmente. El valor predeterminado es `localhost`. Si quiere ejecutar la CLI de AWS SAM en un contenedor Docker en macOS, puede especificar `host.docker.internal`. Si desea ejecutar el contenedor en un host diferente al de AWS SAMCLI, puede especificar la dirección IP del host remoto.

`--container-host-interface` *TEXT*

La dirección IP de la interfaz de red del host a la que deben enlazarse los puertos del contenedor. El valor predeterminado es `127.0.0.1`. Utiliza `0.0.0.0` para enlazarse a todas las interfaces.

`--container-dns` *TEXT*

Especifica un servidor DNS personalizado para el contenedor de Docker. Este parámetro se puede pasar varias veces para especificar varios servidores DNS.

Example

Ejemplo: `--container-dns 8.8.8.8`

`--debug`

Activa el registro de depuración para imprimir el mensaje de depuración que genera la CLI de AWS SAM y muestra las marcas temporales.

`--debug-args` *TEXT*

Los argumentos adicionales que se pasarán al depurador.

--debug-function

Opcional. Especifica la función de eventos de Lambda a la que se aplicarán las opciones de depuración cuando --warm-containers se especifique. Este parámetro se aplica a --debug-port, --debugger-path y --debug-args.

--debug-port, -d *TEXT*

Cuando se especifica, inicia el contenedor de funciones de Lambda en modo de depuración y expone este puerto en el host local.

--debugger-path *TEXT*

La ruta del host a un depurador que se montará en el contenedor de Lambda.

--docker-network *TEXT*

El nombre o el ID de una red de Docker existente a la que deben conectarse los contenedores de Lambda y Docker, junto con la red en modo puente predeterminada. Si esto no se especifica, los contenedores de Lambda se conectan solo a la red Docker en modo puente predeterminada.

--docker-volume-basedir, -v *TEXT*

La ubicación del directorio base en el que se encuentra el AWS SAM archivo. Si Docker se ejecuta en una máquina remota, debe montar la ruta en la que se encuentra el AWS SAM archivo en la máquina Docker y modificar este valor para que coincida con la máquina remota.

--env-vars, -n *PATH*

El JSON o el .env archivo que contiene los valores de las variables de entorno de la función Lambda. El formato del archivo se detecta automáticamente.

--force-image-build

Especifica si la CLI debe reconstruir la imagen utilizada para invocar funciones con capas.

--help

Muestra este mensaje y sale.

--hook-name *TEXT*

El nombre del enlace que se utiliza para ampliar la funcionalidad de la CLI de AWS SAM.

Valores aceptados: terraform.

`--host` *TEXT*

El nombre de host local o la dirección IP a la que se va a vincular (predeterminado: «127.0.0.1»).

`--invoke-image` *TEXT*

El URI de la imagen del contenedor que quiere usar para la invocación de la función local.

Para las funciones Lambda con `PackageType: Zip`, esto anula la imagen base del tiempo de ejecución. De forma predeterminada, AWS SAM extrae la imagen del contenedor de Amazon ECR Public. Utiliza esta opción para extraer la imagen desde otra ubicación.

En el caso de las funciones Lambda con `PackageType: Image`, esto anula la plantilla `ImageUri` creada. Utilice esta opción para invocar localmente una imagen de aplicación diferente sin modificar la plantilla.

Puedes especificar esta opción varias veces. Cada instancia de esta opción puede tomar una cadena o un par clave-valor. Si especificas una cadena, será el URI de la imagen del contenedor que se usará en todas las funciones de la aplicación. Por ejemplo, `sam local start-lambda --invoke-image public.ecr.aws/sam/emu-python3.13`. Si especificas un par clave-valor, la clave es el nombre del recurso y el valor es el URI de la imagen del contenedor que se va a usar para ese recurso. Por ejemplo, `sam local start-lambda --invoke-image public.ecr.aws/sam/emu-python3.13 --invoke-image Function1=amazon/aws-sam-cli-emulation-image-python3.13`. Con los pares clave-valor, puede especificar distintas imágenes de contenedor para distintos recursos.

`--layer-cache-basedir` *DIRECTORY*

Especifica la ubicación basada en el lugar en el que se descargan las capas que utiliza la plantilla.

`--log-file, -l` *TEXT*

El archivo de registro al que se van a enviar los registros de tiempo de ejecución.

`--no-memory-limit`

Elimina la limitación de memoria en el contenedor durante la invocación local, incluso cuando la memoria está configurada en la AWS SAM plantilla.

`--no-watch`

Desactiva la visualización de archivos cuando se usa con. `--warm-containers` Cuando se especifica, AWS SAMCLI no supervisa el código fuente o la plantilla para detectar cambios, y

los contenedores calientes no se reinician automáticamente cuando se modifica el código. Para detectar los cambios locales, detenga el comando y vuelva a ejecutarlo.

Utilice esta opción cuando la supervisión de archivos provoque una I/O sobrecarga o sobrecarga de la CPU, como en monorepos de gran tamaño, entornos con análisis de software antivirus o proyectos en los que varias funciones de Lambda comparten un `CodeUri`.

Esta opción se ignora si no `--warm-containers` se especifica.

`--parameter-overrides`

Cadena que contiene anulaciones de CloudFormation parámetros codificadas como pares clave-valor. Usa el mismo formato que `()`. AWS Command Line Interface AWS CLI El formato de la CLI de AWS SAM es de palabras clave y de valor explícitas, y cada modificación está separada por un espacio. A continuación, se incluyen dos ejemplos:

- `--parameter-overrides ParameterKey=hello,ParameterValue=world`
- `--parameter-overrides ParameterKey=hello,ParameterValue=world
ParameterKey=example1,ParameterValue=example2
ParameterKey=apple,ParameterValue=banana`

`--port, -p INTEGER`

El número de puerto local en el que se va a escuchar (valor predeterminado: «3001»).

`--profile TEXT`

El perfil específico del archivo de credenciales que recibe AWS las credenciales.

`--region TEXT`

La AWS región en la que se va a realizar el despliegue. Por ejemplo, `us-east-1`.

`--save-params`

Guarde los parámetros que proporcione en la línea de comandos en el archivo AWS SAM de configuración.

`--shutdown`

Emula un evento de apagado una vez finalizada la invocación, para probar la forma en que la extensión gestiona el comportamiento del apagado.

`--skip-prepare-infra`

Omite la fase de preparación si no se ha realizado ningún cambio en la infraestructura. Ejecuta con la opción `--hook-name`.

--skip-pull-image

Especifica si la CLI debe omitir la extracción de la última imagen de Docker para el tiempo de ejecución de Lambda.

--template, -t *PATH*

El archivo AWS SAM de plantilla.

Note

Si especifica esta opción, AWS SAM carga únicamente la plantilla y los recursos locales a los que apunta. Esta opción no es compatible con --hook-name.

--terraform-plan-file

La ruta relativa o absoluta al archivo del plan local Terraform cuando se utiliza la CLI de AWS SAM con Terraform Cloud. Esta opción requiere que --hook-name esté configurado en terraform.

--warm-containers [*EAGER* | *LAZY*]

Opcional. Especifica cómo la CLI de AWS SAM administra los contenedores para cada función.

Hay dos opciones disponibles:

- EAGER: los contenedores de todas las funciones se cargan al inicio y se conservan entre las invocaciones.
- LAZY: los contenedores solo se cargan cuando se invoca cada función por primera vez. Esos contenedores se conservan para realizar más invocaciones.

Ejemplo

Para ver un ejemplo y un tutorial detallados sobre el uso del subcomando `sam local start-lambda`, consulte [Introducción a las pruebas con sam local start-lambda](#).

registros de spam

Esta página proporciona información de referencia para el AWS Serverless Application Model comando Command Line Interface (AWS SAMCLI) `sam logs`.

Para obtener una introducción al AWS SAMCLI, consulte [¿Qué es el AWS SAM CLI?](#)

El `sam logs` comando busca los registros generados por sus funciones de AWS Lambda eventos.

De uso

```
$ sam logs <options>
```

Opciones

`--config-env` *TEXT*

El nombre del entorno que especifica los valores de los parámetros predeterminados del archivo de configuración que se va a utilizar. El valor por defecto es «predeterminado». Para obtener más información sobre los archivos de configuración, consulta [Archivo de configuración de la CLI de AWS SAM](#).

`--config-file` *PATH*

La ruta y el nombre del archivo de configuración que contiene los valores de parámetros predeterminados que se van a utilizar. El valor predeterminado es «samconfig.toml» en la raíz del directorio del proyecto. Para obtener más información sobre los archivos de configuración, consulta [Archivo de configuración de la CLI de AWS SAM](#).

`--cw-log-group` *LIST*

Incluye los registros de CloudWatch los grupos de registros que especifique. Si especifica esta opción junto con `name`, AWS SAM incluye los registros de los grupos de registros especificados, además de los registros de los recursos nombrados.

`--debug`

Activa el registro de depuración para imprimir el mensaje de depuración que genera la CLI de AWS SAM y muestra las marcas temporales.

`---end-time`, `e` *TEXT*

Recupera los registros hasta este momento. La hora puede estar formada con valores relativos como «Hace 5 minutos», «mañana» o una marca de tiempo con formato como «01-01-2018 10:10:10».

`--filter` *TEXT*

Permite especificar una expresión para encontrar rápidamente registros que coincidan con términos, frases o valores en los eventos de registro. Puede ser una palabra clave simple

(por ejemplo, «error») o un patrón compatible con Amazon CloudWatch Logs. Para obtener información sobre la sintaxis, consulte la [documentación de Amazon CloudWatch Logs](#).

`--help`

Muestra este mensaje y sale.

`--include-traces`

Incluye X-Ray las trazas en la salida del registro.

`--name, -n TEXT`

El nombre del recurso según el cual se van a buscar los registros. Si este recurso forma parte de una CloudFormation pila, puede ser el ID lógico del recurso de la función en la AWS SAM plantilla CloudFormation/. Para proporcionar varios nombres, repita de nuevo el parámetro. Si el recurso está en una pila anidada, el nombre puede ir precedido del nombre de la pila anidada para extraer los registros de ese recurso (). NestedStackLogicalId/ResourceLogicalId Si no se proporciona el nombre del recurso, se escaneará la pila en cuestión y se extraerá la información de registro de todos los recursos compatibles. Si no especificas esta opción, AWS SAM busca los registros de todos los recursos de la pila que especifiques. Los tipos de recurso siguientes son compatibles:

- `AWS::Serverless::Function`
- `AWS::Lambda::Function`
- `AWS::Serverless::Api`
- `AWS::ApiGateway::RestApi`
- `AWS::Serverless::HttpApi`
- `AWS::ApiGatewayV2::Api`
- `AWS::Serverless::StateMachine`
- `AWS::StepFunctions::StateMachine`

`--output TEXT`

Especifica el formato de salida de los registros. Para imprimir registros formateados, especifica `text`. Para imprimir los registros como JSON, especifica `json`.

`--profile TEXT`

El perfil específico del archivo de credenciales que obtiene AWS las credenciales.

`--region` *TEXT*

La AWS región en la que se va a realizar el despliegue. Por ejemplo, us-east-1.

`--save-params`

Guarde los parámetros que proporcione en la línea de comandos en el archivo AWS SAM de configuración.

`--stack-name` *TEXT*

El nombre de la CloudFormation pila de la que forma parte el recurso.

`--start-time, -s` *TEXT*

Recupera los registros que comienzan en este momento. La hora puede estar formada con valores relativos como «Hace 5 minutos», «ayer» o una marca de tiempo con formato como «01-01-2018 10:10:10». El valor predeterminado es «Hace 10 minutos».

`--tail, -t`

Sigue la salida del registro. Esto ignora el argumento de la hora de finalización y continúa buscando los registros a medida que están disponibles.

Ejemplos

Cuando las funciones forman parte de una CloudFormation pila, puede recuperar los registros mediante el ID lógico de la función al especificar el nombre de la pila.

```
$ sam logs -n HelloWorldFunction --stack-name myStack
```

Consulte los registros de un intervalo de tiempo específico mediante las opciones `-s` (`--start-time`) y `-e` (`--end-time`).

```
$ sam logs -n HelloWorldFunction --stack-name myStack -s '10min ago' -e '2min ago'
```

También puede agregar la opción `--tail` para esperar a que lleguen nuevos registros y verlos a medida que lleguen.

```
$ sam logs -n HelloWorldFunction --stack-name myStack --tail
```


Utiliza la opción `--filter` para encontrar rápidamente registros que coincidan con términos, frases o valores de sus eventos de registro.

```
$ sam logs -n HelloWorldFunction --stack-name myStack --filter "error"
```

Vea los registros de un recurso en una pila secundaria.

```
$ sam logs --stack-name myStack -n childStack/HelloWorldFunction
```

Registros finales de todos los recursos compatibles con la aplicación.

```
$ sam logs --stack-name sam-app --tail
```

Obtén los registros de una función de Lambda específica y de la API correspondiente a API Gateway de su aplicación.

```
$ sam logs --stack-name sam-app --name HelloWorldFunction --name HelloWorldRestApi
```

Obtén los registros de todos los recursos admitidos en su aplicación y, además, de los grupos de registros especificados.

```
$ sam logs --cw-log-group /aws/lambda/myfunction-123 --cw-log-group /aws/lambda/myfunction-456
```

sam package

La interfaz de línea de AWS Serverless Application Model comandos (AWS SAM CLI) empaqueta una AWS SAM aplicación.

Este comando crea un `.zip` archivo con el código y las dependencias y lo carga en Amazon Simple Storage Service (Amazon S3). AWS SAM permite el cifrado de todos los archivos almacenados en Amazon S3. A continuación, devuelve una copia de la AWS SAM plantilla y reemplaza las referencias a los artefactos locales por la ubicación de Amazon S3 en la que el comando cargó los artefactos.

De forma predeterminada, cuando utiliza este comando, CLI de AWS SAM asume que su directorio de trabajo actual es el directorio raíz del proyecto. El AWS SAMCLI primero intenta localizar un archivo de plantilla creado con el [compilación de sam](#) comando, ubicado en la `.aws-sam` subcarpeta

y `nombradotemplate.yaml`. A continuación, la CLI de AWS SAM intenta localizar un archivo de plantilla llamado `template.yaml` o `template.yml` en el directorio de trabajo actual. Si especificas la `--template` opción, se AWS SAMCLI anula su comportamiento predeterminado y empaquetará solo esa AWS SAM plantilla y los recursos locales a los que apunte.

Note

[sam deploy](#) ahora ejecuta implícitamente la funcionalidad del `sam package`. Puede utilizar el comando [sam deploy](#) directamente para empaquetar e implementar su aplicación.

De uso

```
$ sam package <arguments> <options>
```

Argumentos

Resource ID (ID de recurso)

El ID de la función de eventos de Lambda que se va a empaquetar.

Este argumento es opcional. Si la aplicación contiene una sola función Lambda, la AWS SAM CLI la empaquetará. Si la aplicación contiene varias funciones, proporciones el ID de la función para empaquetar una sola función.

Valores válidos: el identificador lógico o el ARN del recurso.

Opciones

`--config-env` *TEXT*

El nombre del entorno que especifica los valores de los parámetros predeterminados del archivo de configuración que se va a utilizar. El valor por defecto es «predeterminado». Para obtener más información sobre los archivos de configuración, consulta [Archivo de configuración de la CLI de AWS SAM](#).

`--config-file` *PATH*

La ruta y el nombre del archivo de configuración que contiene los valores de parámetros predeterminados que se van a utilizar. El valor predeterminado es «samconfig.toml» en la raíz

del directorio del proyecto. Para obtener más información sobre los archivos de configuración, consulta [Archivo de configuración de la CLI de AWS SAM](#).

--debug

Activa el registro de depuración para imprimir el mensaje de depuración que genera la CLI de AWS SAM y muestra las marcas temporales.

--force-upload

Anular los archivos existentes en el bucket de Amazon S3. Especificar este indicador para cargar artefactos incluso si coinciden con los artefactos existentes en el bucket de Amazon S3.

--help

Muestra este mensaje y sale.

--image-repository *TEXT*

La URI del repositorio Amazon Elastic Container Registry (Amazon ECR) donde este comando sube la imagen de la función. Necesario para las funciones declaradas con el tipo de paquete Image.

--kms-key-id *TEXT*

El ID de una clave AWS Key Management Service (AWS KMS) que se utiliza para cifrar los artefactos que se encuentran en reposo en el bucket de Amazon S3. Si no se especifica esta opción, AWS SAM utiliza las claves de S3-managed cifrado de Amazon.

--metadata

(Opcional) Un mapa de metadatos para adjuntarlo a todos los artefactos a los que se hace referencia en la plantilla.

--no-progressbar

No muestre una barra de progreso al cargar artefactos en Amazon S3.

--output-template-file *PATH*

La ruta al archivo donde el comando escribe la plantilla empaquetada. Si especificas una ruta, el comando escribe la plantilla en la salida estándar.

--profile *TEXT*

El perfil específico del archivo de credenciales que recibe AWS las credenciales.

`--region` *TEXT*

La AWS región en la que se va a realizar el despliegue. Por ejemplo, us-east-1.

`--resolve-s3`

Creación automática de un bucket de Amazon S3 para empaquetar. Si especificas las opciones `--s3-bucket` y `--resolve-s3`, entonces se producirá un error.

`--s3-bucket` *TEXT*

El nombre del bucket de Amazon S3 donde este comando carga el artefacto. Si el artefacto tiene más de 51 200 bytes, necesitará la opción `--s3-bucket` o `--resolve-s3`. Si especificas las opciones `--s3-bucket` y `--resolve-s3`, entonces se producirá un error.

`--s3-prefix` *TEXT*

Prefijo agregado al nombre de los artefactos que se cargan en el bucket de Amazon S3. El nombre del prefijo es un nombre de ruta (nombre de la carpeta) del bucket de Amazon S3. Esto solo se aplica a las funciones declaradas con el tipo de paquete Zip.

`--save-params`

Guarde los parámetros que proporcione en la línea de comandos en el archivo AWS SAM de configuración.

`--signing-profiles` *LIST*

(Opcional) La lista de perfiles de firma con los que firmar los paquetes de implementación. Este parámetro toma una lista de pares clave-valor, donde la clave es el nombre de la función o capa que se va a firmar y el valor es el perfil de firma, con un propietario del perfil opcional delimitado con `:`. Por ejemplo, `FunctionNameToSign=SigningProfileName1`
`LayerNameToSign=SigningProfileName2:SigningProfileOwner`.

`--template-file`, `--template`, `-t` *PATH*

La ruta y el nombre del archivo donde se encuentra la AWS SAM plantilla.

 Note

Si especifica esta opción, AWS SAM empaqueta solo la plantilla y los recursos locales a los que apunta.

`--use-json`

Genera JSON para la CloudFormation plantilla. YAML se utiliza de forma predeterminada.

Ejemplo

El siguiente ejemplo crea y empaqueta artefactos para una función y CodeDeploy aplicaciones de Lambda. Los artefactos se cargan en un bucket de Amazon S3. El resultado del comando es un nuevo archivo llamado `package.yml`.

```
$ sam package \
  --template-file template.yml \
  --output-template-file package.yml \
  --s3-bucket amzn-s3-demo-bucket
```

sam pipeline bootstrap

Esta página proporciona información de referencia para el AWS Serverless Application Model `sam local pipeline bootstrap` subcomando de la Interfaz de línea de comandos (AWS SAMCLI).

Para obtener una introducción al AWS SAMCLI, consulte [¿Qué es el AWS SAM CLI?](#)

El `sam pipeline bootstrap` subcomando genera los recursos de AWS infraestructura necesarios para conectarse al CI/CD sistema. Este paso debe ejecutarse en cada etapa de implementación de la canalización antes de ejecutar el comando `sam pipeline init`.

Este subcomando configura los siguientes recursos de AWS infraestructura:

- Opción de configurar los permisos de canalización mediante:
 - Un usuario de Pipeline IAM con un identificador de clave de acceso y unas credenciales de acceso con clave secreta que se compartirán con el CI/CD sistema.

Note

Recomendamos rotar periódicamente las claves de acceso de forma regular. Para obtener más información, consulta [Rotar las claves de acceso periódicamente para casos de uso que requieran credenciales de larga duración](#) en la Guía del usuario de IAM.

- CI/CD Plataformas compatibles a través del OIDC. Para obtener una introducción sobre el uso de OIDC con la canalización AWS SAM , vaya a [Cómo utilizar la autenticación OIDC con canalizaciones AWS SAM](#).
- Función de IAM de CloudFormation ejecución que asume el despliegue de CloudFormation la aplicación. AWS SAM
- Un depósito de Amazon S3 para guardar los AWS SAM artefactos.
- Opcionalmente, un repositorio de imágenes de Amazon ECR para almacenar paquetes de implementación de Lambda con imágenes de contenedor (si tiene un recurso del tipo de paquete Image).

De uso

```
$ sam pipeline bootstrap <options>
```

Opciones

`--bitbucket-repo-uuid` *TEXT*

El UUID del repositorio de Bitbucket. Esta opción es específica del uso de Bitbucket OIDC para permisos.

Note

Este valor se encuentra en <https://bitbucket.org/workspace//repositoryadmin/addon/admin/pipelines/openid-connect>

`--bucket` *TEXT*

El ARN del depósito de Amazon S3 que contiene los AWS SAM artefactos.

`--cicd-provider` *TEXT*

La CI/CD plataforma para la AWS SAM canalización.

`--cloudformation-execution-role` *TEXT*

El ARN de la función de IAM que debe asumir CloudFormation al implementar la pila de aplicaciones. indícalo solo si quiere utilizar su propio rol. De lo contrario, el comando creará un nuevo rol.

`--config-env` *TEXT*

El nombre del entorno que especifica los valores de los parámetros predeterminados del archivo de configuración que se va a utilizar. El valor predeterminado es **default**. Para obtener más información sobre los archivos de configuración, consulta [Archivo de configuración de la CLI de AWS SAM](#).

`--config-file` *PATH*

La ruta y el nombre del archivo de configuración que contiene los valores de los parámetros predeterminados que se van a utilizar. El valor por defecto es `samconfig.toml` en la raíz del directorio del proyecto. Para obtener más información sobre los archivos de configuración, consulta [Archivo de configuración de la CLI de AWS SAM](#).

`--confirm-changeset` | `--no-confirm-changeset`

Permite solicitar la confirmación de la implementación de los recursos.

`--create-image-repository` | `--no-create-image-repository`

Especifica si quiere crear un repositorio de imágenes de Amazon ECR si no se proporciona ninguno. El repositorio de Amazon ECR contiene las imágenes de contenedor de las funciones de eventos de Lambda, o las capas que tienen un tipo de paquete de. Image El valor predeterminado es `--no-create-image-repository`.

`--debug`

Activa el registro de depuración e imprime los mensajes de depuración que genera la CLI de AWS SAM, además de mostrar las marcas de tiempo.

`--deployment-branch` *TEXT*

Nombre de la rama desde la que se realizarán las implementaciones. Esta opción es específica del uso de GitHub Actions OIDC para los permisos.

`--github-org` *TEXT*

La GitHub organización a la que pertenece el repositorio. Si no existe ninguna organización, escriba el nombre de usuario del propietario del repositorio. Esta opción es específica del uso de GitHub Actions OIDC para los permisos.

`--github-repo` *TEXT*

Nombre del GitHub repositorio desde el que se realizarán las implementaciones. Esta opción es específica del uso de GitHub Actions OIDC para los permisos.

`--gitlab-group` *TEXT*

El GitLab grupo al que pertenece el repositorio. Esta opción es específica del uso de GitLab OIDC para los permisos.

`--gitlab-project` *TEXT*

El nombre del GitLab proyecto. Esta opción es específica del uso de GitLab OIDC para los permisos.

`--help`, `-h`

Muestra este mensaje y sale.

`--image-repository` *TEXT*

El ARN de un repositorio de imágenes de Amazon ECR en el que se encuentran las imágenes de contenedor de las funciones de Lambda o las capas que tienen un tipo de paquete de Image. Si se proporcionan, se ignora la opción `--create-image-repository`. Si no se proporciona y `--create-image-repository` se especifica, el comando crea una.

`--interactive` | `--no-interactive`

Se deshabilita la solicitud interactiva de los parámetros de arranque y devuelve un error si falta alguno de los parámetros necesarios. El valor predeterminado es `--interactive`. Para este comando, `--stage` es el único parámetro requerido.

 Note

Si `--no-interactive` se especifica junto con `--use-oidc-provider`, se deben incluir todos los parámetros necesarios de su proveedor de OIDC.

`--oidc-client-id` *TEXT*

El ID de cliente configurado para su uso con su proveedor de OIDC.

`--oidc-provider` [*github-actions* | *gitlab* | *bitbucket-pipelines*]

Nombre del CI/CD proveedor que se utilizará para los permisos de OIDC. GitLab GitHub, y Bitbucket son compatibles.

`--oidc-provider-url` *TEXT*

La dirección URL del emisor del proveedor OIDC. El valor debe empezar por **https://**.

`--permissions-provider` *[oidc | iam]*

Elija un proveedor de permisos para que asuma la función de ejecución de la canalización. El valor predeterminado es **iam**.

`--pipeline-execution-role` *TEXT*

El ARN del rol de IAM que asumirá el usuario de la canalización y que usará en esta etapa. Indíquelo solo si quiere utilizar su propio rol. Si no se proporciona, este comando creará un nuevo rol.

`--pipeline-user` *TEXT*

El nombre de recurso de Amazon (ARN) del usuario de IAM que comparte con el sistema su ID de clave de acceso y su clave de acceso secreta. CI/CD Se utiliza para conceder permiso a este usuario de IAM para acceder a la cuenta correspondiente. AWS Si no se proporciona, el comando creará un usuario de IAM junto con el ID de clave de acceso y las credenciales de clave de acceso secreta.

`--profile` *TEXT*

El perfil específico del archivo de credenciales que recibe AWS las credenciales.

`--region` *TEXT*

La AWS región en la que se va a realizar el despliegue. Por ejemplo, `us-east-1`.

`--save-params`

Guarde los parámetros que proporcione en la línea de comandos en el archivo AWS SAM de configuración.

`--stage` *TEXT*

El nombre de la fase de implementación correspondiente. Se utiliza como sufijo para los recursos de AWS infraestructura creados.

Resolución de problemas

Error: falta el parámetro obligatorio

Si `--no-interactive` se especifica junto con `--use-oidc-provider` y ninguno de los parámetros requeridos se proporciona, se mostrará este mensaje de error junto con una descripción de los parámetros que faltan.

Ejemplo

El siguiente ejemplo crea los AWS recursos necesarios para crear el CI/CD sistema, activa el registro de depuración e imprime los mensajes de depuración generados por AWS SAMCLI: usa un evento generado para pruebas locales mediante el uso de un `s3.json` evento para invocar una función Lambda localmente

```
$ sam pipeline bootstrap --debug
```

sam pipeline init

Esta página proporciona información de referencia para el AWS Serverless Application Model `sam pipeline init` subcomando de la Interfaz de línea de comandos (AWS SAMCLI).

Para obtener una introducción al AWS SAMCLI, consulte [¿Qué es el AWS SAM CLI?](#)

El `sam pipeline init` subcomando genera un archivo de configuración de canalización que el CI/CD sistema puede utilizar para implementar aplicaciones sin servidor. AWS SAM

Antes de usar `sam pipeline init`, debe iniciar los recursos necesarios para cada etapa de la canalización. Puede hacer esto ejecutando `sam pipeline init --bootstrap` para que lo guíen a través del proceso de instalación y generación de archivos de configuración, o consulta los recursos que haya creado previamente con el comando `sam pipeline bootstrap`.

De uso

```
$ sam pipeline init <options>
```

Opciones

`--bootstrap`

Habilite el modo interactivo que guíe al usuario a través de la creación de los recursos de AWS infraestructura necesarios.

`--config-env` *TEXT*

El nombre del entorno que especifica los valores de los parámetros predeterminados del archivo de configuración que se va a utilizar. El valor predeterminado es `default`. Para obtener más

información sobre los archivos de configuración, consulta [Archivo de configuración de la CLI de AWS SAM](#).

`--config-file` *TEXT*

La ruta y el nombre del archivo de configuración que contiene los valores de parámetros predeterminados que se van a utilizar. El valor predeterminado es `samconfig.toml` en el directorio raíz del proyecto. Para obtener más información sobre los archivos de configuración, consulta [Archivo de configuración de la CLI de AWS SAM](#).

`--debug`

Activa el registro de depuración para imprimir los mensajes de depuración que genera la AWS SAM de CLI y para mostrar las marcas temporales.

`--help`, `-h`

Muestra este mensaje y sale.

`--save-params`

Guarde los parámetros que proporcione en la línea de comandos en el archivo AWS SAM de configuración.

Ejemplo

En el siguiente ejemplo, se muestra cómo utilizar la `--bootstrap` opción para realizar un recorrido por un modo interactivo que le guiará por la creación de los recursos de AWS infraestructura necesarios:

```
$ sam pipeline init --bootstrap
```

sam publish

Esta página proporciona información de referencia para el AWS Serverless Application Model comando Command Line Interface (AWS SAMCLI) `sam publish`.

Para obtener una introducción al AWS SAMCLI, consulte [¿Qué es el AWS SAM CLI?](#)

El `sam publish` comando publica una AWS SAM aplicación en AWS Serverless Application Repository. Este comando toma una AWS SAM plantilla empaquetada y publica la aplicación en la AWS región especificada.

El `aws sam publish` comando espera que la AWS SAM plantilla incluya una Metadata sección que contenga los metadatos de la aplicación necesarios para la publicación. En la sección Metadata, las propiedades `LicenseUrl` y `ReadmeUrl` deben hacer referencia a buckets de Amazon Simple Storage Service (Amazon S3), no a archivos locales. Para obtener más información sobre la Metadata sección de la AWS SAM plantilla, consulte [Publicación de la aplicación con la de AWS SAM CLI](#).

De forma predeterminada, `aws sam publish` crea la aplicación como privada. Antes de que otras cuentas AWS puedan ver e implementar su aplicación, debe compartirla. Para obtener información sobre cómo compartir aplicaciones, consulta [los ejemplos de políticas basadas en recursos de AWS Serverless Application Repository](#) en la Guía para desarrolladores de AWS Serverless Application Repository .

Note

Por el momento, `aws sam publish` no admite la publicación de aplicaciones anidadas especificadas localmente. Si la aplicación contiene aplicaciones anidadas, debe publicarlas por separado AWS Serverless Application Repository antes de publicar la aplicación principal.

De uso

```
$ aws sam publish <options>
```

Opciones

`--config-env` *TEXT*

El nombre del entorno que especifica los valores de los parámetros predeterminados del archivo de configuración que se va a utilizar. El valor por defecto es «predeterminado». Para obtener más información sobre los archivos de configuración, consulta [Archivo de configuración de la CLI de AWS SAM](#).

`--config-file` *PATH*

La ruta y el nombre del archivo de configuración que contiene los valores de parámetros predeterminados que se van a utilizar. El valor por defecto es «samconfig.toml» en la raíz

del directorio del proyecto. Para obtener más información sobre los archivos de configuración, consulta [Archivo de configuración de la CLI de AWS SAM](#).

`--debug`

Activa el registro de depuración para imprimir los mensajes de depuración que genera la AWS SAM de CLI y para mostrar las marcas temporales.

`--help`

Muestra este mensaje y sale.

`--profile` *TEXT*

El perfil específico del archivo de credenciales que recibe AWS las credenciales.

`--region` *TEXT*

La AWS región en la que se va a realizar el despliegue. Por ejemplo, us-east-1.

`--save-params`

Guarde los parámetros que proporcione en la línea de comandos en el archivo AWS SAM de configuración.

`--semantic-version` *TEXT*

(Opcional) Utiliza esta opción para proporcionar una versión semántica de la aplicación que anule la propiedad SemanticVersion de la sección Metadata del archivo de plantilla. Para obtener más información sobre el control de versiones semántico, consulta la [especificación del control de versiones semántico](#).

`--template, -t` *PATH*

La ruta del archivo de AWS SAM plantilla[default: `template.[yaml|yml]`].

Ejemplos

Para publicar una aplicación:

```
$ sam publish --template packaged.yaml --region us-east-1
```

sam remote callback

Esta página proporciona información de referencia para el AWS Serverless Application Model comando Command Line Interface (AWS SAMCLI)sam remote callback.

Para obtener una introducción al AWS SAMCLI, consulte [¿Qué es el AWS SAM CLI?](#)

El `sam remote callback` comando le permite enviar llamadas a ejecuciones remotas de funciones duraderas que se ejecutan en AWS Lambda.

 Note

Estos comandos requieren acceso a AWS las credenciales.

De uso

```
$ sam remote callback <subcommand> <options>
```

Opciones

--help, -h

Muestre este mensaje y salga de él.

Subcomandos

succeed

Envíe una llamada de respuesta correcta a una ejecución remota de una función duradera. Para obtener más información, consulte [sam remote callback succeed](#).

fail

Envía un error de devolución de llamada a una ejecución remota de una función duradera. Para obtener más información, consulte [sam remote callback fail](#).

heartbeat

Envía una llamada de retorno a una ejecución remota de una función duradera. Para obtener más información, consulte [sam remote callback heartbeat](#).

sam remote callback succeed

Envíe una llamada correcta a una ejecución remota de una función duradera.

 Note

Este comando requiere acceso a AWS las credenciales.

De uso

```
$ sam remote callback succeed CALLBACK_ID [OPTIONS]
```

Argumentos necesarios

CALLBACK_ID

El identificador único de la devolución de llamada a la que se debe enviar una respuesta correcta.

Opciones

-r, --result TEXT

Resultado correcto: carga útil en forma de cadena.

--region TEXT

Defina la AWS región del servicio (por ejemplo,us-east-1).

--profile TEXT

Seleccione un perfil específico de su archivo de credenciales para obtener AWS las credenciales.

--config-env TEXT

Nombre del entorno que especifica los valores de los parámetros predeterminados en el archivo de configuración. Valor predeterminado: default

--config-file TEXT

Archivo de configuración que contiene los valores de los parámetros predeterminados. Valor predeterminado: samconfig.toml

--save-params

Guarde los parámetros proporcionados a través de la línea de comandos en el archivo de configuración.

`--beta-features / --no-beta-features`

Activar/desactivar las funciones beta.

`--debug`

Active el registro de depuración para imprimir los mensajes de depuración generados por la AWS SAM CLI y mostrar las marcas de tiempo.

`-h, --help`

Muestra este mensaje y sale.

Ejemplos

Enviar una llamada de éxito sin resultado:

```
$ sam remote callback succeed my-callback-id
```

Envía una llamada de confirmación con el resultado:

```
$ sam remote callback succeed my-callback-id --result 'Task completed successfully'
```

sam remote callback fail

Envía una llamada fallida a una ejecución remota de una función duradera.

Note

Este comando requiere acceso a AWS las credenciales.

De uso

```
$ sam remote callback fail CALLBACK_ID [OPTIONS]
```

Argumentos necesarios

CALLBACK_ID

El identificador único de la devolución de llamada a la que se va a enviar una respuesta de error.

Opciones

`--error-data TEXT`

Datos de error adicionales.

`--stack-trace TEXT`

Apila las entradas de rastreo. Esta opción se puede especificar varias veces.

`--error-type TEXT`

Tipo de error.

`--error-message TEXT`

Mensaje de error detallado.

`--region TEXT`

Defina la AWS región del servicio (por ejemplo, `us-east-1`).

`--profile TEXT`

Seleccione un perfil específico de su archivo de credenciales para obtener AWS las credenciales.

`--config-env TEXT`

Nombre del entorno que especifica los valores de los parámetros predeterminados en el archivo de configuración. Valor predeterminado: `default`

`--config-file TEXT`

Archivo de configuración que contiene los valores de los parámetros predeterminados. Valor predeterminado: `samconfig.toml`

`--save-params`

Guarde los parámetros proporcionados a través de la línea de comandos en el archivo de configuración.

`--beta-features / --no-beta-features`

Activar/desactivar las funciones beta.

`--debug`

Active el registro de depuración para imprimir los mensajes de depuración generados por la AWS SAM CLI y mostrar las marcas de tiempo.

-h, --help

Muestra este mensaje y sale.

Ejemplos

Envía una llamada de error sin parámetros:

```
$ sam remote callback fail my-callback-id
```

Envía una llamada de error con un mensaje de error:

```
$ sam remote callback fail my-callback-id --error-message 'Task failed'
```

Envía una llamada de error con todos los parámetros:

```
$ sam remote callback fail my-callback-id --error-message 'Task failed' --error-type 'ValidationError' --stack-trace 'at line 42' --error-data '{"code": 500}'
```

sam remote callback heartbeat

Envíe un latido de llamada a una ejecución remota de una función duradera.

Note

Este comando requiere acceso a AWS las credenciales.

De uso

```
$ sam remote callback heartbeat CALLBACK_ID [OPTIONS]
```

Argumentos necesarios

CALLBACK_ID

El identificador único de la devolución de llamada a la que se va a enviar un latido.

Opciones

`--region TEXT`

Defina la AWS región del servicio (por ejemplo, `us-east-1`).

`--profile TEXT`

Seleccione un perfil específico de su archivo de credenciales para obtener AWS las credenciales.

`--config-env TEXT`

Nombre del entorno que especifica los valores de los parámetros predeterminados en el archivo de configuración. Valor predeterminado: `default`

`--config-file TEXT`

Archivo de configuración que contiene los valores de los parámetros predeterminados. Valor predeterminado: `samconfig.toml`

`--save-params`

Guarde los parámetros proporcionados a través de la línea de comandos en el archivo de configuración.

`--beta-features / --no-beta-features`

Activar/desactivar las funciones beta.

`--debug`

Active el registro de depuración para imprimir los mensajes de depuración generados por la AWS SAM CLI y mostrar las marcas de tiempo.

`-h, --help`

Muestra este mensaje y sale.

Ejemplos

Enviar una llamada de latido:

```
$ sam remote callback heartbeat my-callback-id
```

sam remote execution

Esta página proporciona información de referencia para el AWS Serverless Application Model comando Command Line Interface (AWS SAMCLI) `sam remote execution`.

Para obtener una introducción al AWS SAMCLI, consulte [¿Qué es el AWS SAM CLI?](#)

El `sam remote execution` comando le permite gestionar e inspeccionar las ejecuciones remotas de funciones duraderas que se ejecutan en AWS Lambda.

Note

Estos comandos requieren acceso a AWS las credenciales.

De uso

```
$ sam remote execution <subcommand> <options>
```

Opciones

`--help`, `-h`

Muestre este mensaje y salga de él.

Subcomandos

get

Obtenga detalles de una ejecución duradera. Para obtener más información, consulte [sam remote execution get](#).

history

Obtenga el historial de ejecución de una función duradera. Para obtener más información, consulte [sam remote execution history](#).

stop

Detenga la ejecución duradera de una función. Para obtener más información, consulte [sam remote execution stop](#).

sam remote execution get

Recupere los detalles de la ejecución de una función duradera específica de AWS Lambda.

El ARN de ejecución debe tener el formato devuelto por Invoke o encontrarse en los registros de ejecución.

Note

Este comando requiere acceso a AWS las credenciales.

De uso

```
$ sam remote execution get DURABLE_EXECUTION_ARN [OPTIONS]
```

Argumentos necesarios

DURABLE_EXECUTION_ARN

El nombre del recurso de Amazon (ARN) de la ejecución de la función duradera de la que se van a recuperar los detalles.

Opciones

--format [summary|json]

Formato de salida. Valor predeterminado: summary

--region TEXT

Defina la AWS región del servicio (por ejemplo,us-east-1).

--profile TEXT

Seleccione un perfil específico de su archivo de credenciales para obtener AWS las credenciales.

--config-env TEXT

Nombre del entorno que especifica los valores de los parámetros predeterminados en el archivo de configuración. Valor predeterminado: default

--config-file TEXT

Archivo de configuración que contiene los valores de los parámetros predeterminados. Valor predeterminado: `samconfig.toml`

--save-params

Guarde los parámetros proporcionados a través de la línea de comandos en el archivo de configuración.

--beta-features / --no-beta-features

Activar/desactivar las funciones beta.

--debug

Active el registro de depuración para imprimir los mensajes de depuración generados por la AWS SAM CLI y mostrar las marcas de tiempo.

-h, --help

Muestra este mensaje y sale.

Ejemplos

Obtenga los detalles de la ejecución con el formato de resumen predeterminado:

```
$ sam remote execution get 'arn:aws:lambda:us-east-1:123456789012:function:my-function:
$LATEST/durable-execution/c63eec67-3415-4eb4-a495-116aa3a86278/1d454231-a3ad-3694-aa03-
c917c175db55'
```

Obtenga los detalles de la ejecución en formato JSON:

```
$ sam remote execution get 'arn:aws:lambda:us-east-1:123456789012:function:my-function:
$LATEST/durable-execution/c63eec67-3415-4eb4-a495-116aa3a86278/1d454231-a3ad-3694-aa03-
c917c175db55' --format json
```

Note

Debe asegurarse de que los caracteres de control del ARN de ejecución \$ se escapen correctamente cuando utilice comandos de shell.

sam remote execution history

Recupere el historial de ejecución de una función duradera específica de AWS Lambda.

El ARN de ejecución debe tener el formato devuelto StartDurableExecution o encontrado en los registros de ejecución.

Note

Este comando requiere acceso a AWS las credenciales.

De uso

```
$ sam remote execution history DURABLE_EXECUTION_ARN [OPTIONS]
```

Argumentos necesarios

DURABLE_EXECUTION_ARN

El nombre del recurso de Amazon (ARN) de la ejecución duradera de la función para recuperar el historial.

Opciones

--format [table|json]

Formato de salida. Valor predeterminado: table

--region TEXT

Defina la AWS región del servicio (por ejemplo,us-east-1).

--profile TEXT

Seleccione un perfil específico de su archivo de credenciales para obtener AWS las credenciales.

--config-env TEXT

Nombre del entorno que especifica los valores de los parámetros predeterminados en el archivo de configuración. Valor predeterminado: default

--config-file TEXT

Archivo de configuración que contiene los valores de los parámetros predeterminados. Valor predeterminado: `samconfig.toml`

--save-params

Guarde los parámetros proporcionados a través de la línea de comandos en el archivo de configuración.

--beta-features / --no-beta-features

Activar/desactivar las funciones beta.

--debug

Active el registro de depuración para imprimir los mensajes de depuración generados por la AWS SAM CLI y mostrar las marcas de tiempo.

-h, --help

Muestra este mensaje y sale.

Ejemplos

Obtenga el historial de ejecuciones con el formato de tabla predeterminado:

```
$ sam remote execution history arn:aws:lambda:us-east-1:123456789012:function:MyFunction:$LATEST/durable-execution/my-execution-name/my-execution-id
```

Obtenga el historial de ejecuciones en formato JSON:

```
$ sam remote execution history arn:aws:lambda:us-east-1:123456789012:function:MyFunction:$LATEST/durable-execution/my-execution-name/my-execution-id --format json
```

sam remote execution stop

Detenga la ejecución duradera de una función en AWS Lambda.

El ARN de ejecución debe tener el formato devuelto `StartDurableExecution` o encontrado en los registros de ejecución.

 Note

Este comando requiere acceso a AWS las credenciales.

De uso

```
$ sam remote execution stop DURABLE_EXECUTION_ARN [OPTIONS]
```

Argumentos necesarios

DURABLE_EXECUTION_ARN

El nombre del recurso de Amazon (ARN) de la ejecución duradera de la función que se va a detener.

Opciones

--error-message TEXT

Mensaje de error asociado a la ejecución detenida.

--error-type TEXT

Tipo de error que se va a asociar a la ejecución detenida.

--error-data TEXT

Datos de error que se van a asociar a la ejecución detenida.

--stack-trace TEXT

Apila las entradas de rastreo. Esta opción se puede especificar varias veces.

--region TEXT

Defina la AWS región del servicio (por ejemplo, `us-east-1`).

--profile TEXT

Seleccione un perfil específico de su archivo de credenciales para obtener AWS las credenciales.

--config-env TEXT

Nombre del entorno que especifica los valores de los parámetros predeterminados en el archivo de configuración. Valor predeterminado: `default`

--config-file TEXT

Archivo de configuración que contiene los valores de los parámetros predeterminados. Valor predeterminado: `samconfig.toml`

--save-params

Guarde los parámetros proporcionados a través de la línea de comandos en el archivo de configuración.

--beta-features / --no-beta-features

Activar/desactivar las funciones beta.

--debug

Active el registro de depuración para imprimir los mensajes de depuración generados por la AWS SAM CLI y mostrar las marcas de tiempo.

-h, --help

Muestra este mensaje y sale.

Ejemplos

Detenga la ejecución sin detalles del error:

```
$ sam remote execution stop arn:aws:lambda:us-east-1:123456789012:function:MyFunction:
$LATEST/durable-execution/my-execution-name/my-execution-id
```

Detenga la ejecución con un mensaje de error y escriba:

```
$ sam remote execution stop arn:aws:lambda:us-east-1:123456789012:function:MyFunction:
$LATEST/durable-execution/my-execution-name/my-execution-id --error-message "Execution
cancelled" --error-type "UserCancellation"
```

Detenga la ejecución con todos los detalles del error y el seguimiento de la pila:

```
$ sam remote execution stop arn:aws:lambda:us-east-1:123456789012:function:MyFunction:
$LATEST/durable-execution/my-execution-name/my-execution-id --error-message "Task
failed" --error-type "TaskFailure" --error-data '{"reason":"timeout"}' --stack-trace
"at function1()" --stack-trace "at function2()"
```

invocación remota de sam

Esta página proporciona información de referencia para el AWS Serverless Application Model comando Command Line Interface (AWS SAMCLI) `sam remote invoke`.

- Para obtener una introducción al AWS SAMCLI, consulte [¿Qué es el AWS SAM CLI?](#)
- Para obtener documentación sobre el uso del AWS SAMCLI `sam remote invoke` comando, consulte [Introducción a las pruebas en la nube con invocación remota de sam](#).

El comando `sam remote invoke` invoca los recursos compatibles en Nube de AWS.

De uso

```
$ sam remote invoke <arguments> <options>
```

Argumentos

Resource ID (ID de recurso)

El ID del recurso admitido a invocar.

Este argumento acepta los valores siguientes:

- El Nombre de recurso de Amazon (ARN): el ARN del recurso.

Tip

Usa `sam list stack-outputs --stack-name <stack-name>` para obtener el ARN de sus recursos.

- ID lógico: el ID lógico del recurso. También debe proporcionar el nombre de la AWS CloudFormation pila mediante la `--stack-name` opción.
- ID físico: el ID físico del recurso. Este ID se crea al implementar un recurso utilizando CloudFormation.

Tip

Usa `sam list resources --stack-name <stack-name>` para obtener el ID físico de sus recursos.

Cuando proporciona un ARN o una identificación física:

Si proporcionas un ARN o un ID físico, no proporcionas un nombre de pila. Cuando el nombre de la pila se proporciona mediante la `--stack-name` opción, o cuando el nombre de la pila se define en el archivo de configuración, AWS SAM CLI procesará automáticamente el ID de recurso como un valor de ID lógico de la CloudFormation pila.

Cuando no proporciona un ID de recurso:

Si no proporciona un ID de recurso, pero sí proporciona un nombre de pila con la `--stack-name` opción, la AWS SAM CLI intentará invocar automáticamente un recurso de la CloudFormation pila mediante la siguiente lógica:

1. AWS SAM CLI identificará los tipos de recursos en el siguiente orden y pasará al siguiente paso una vez que el tipo de recurso se encuentre en la pila:
 - a. Lambda
 - b. Step Functions
 - c. Amazon SQS
 - d. Kinesis Data Streams
2. Si el tipo de recurso tiene un solo recurso en tu pila, lo AWS SAM CLI invocará. Si hay varios recursos del mismo tipo en tu pila, se AWS SAM CLI mostrará un error.

Los siguientes son ejemplos de lo que AWS SAM CLI hará:

- Pila que contiene dos funciones de eventos de Lambda y una cola de Amazon SQS: AWS SAM CLI localizará el tipo de recurso de Lambda y devolverá un error, ya que la pila contiene más de una función de Lambda.
- Pila que contiene una función Lambda y dos aplicaciones de Amazon Kinesis Data Streams: AWS SAM CLI localizará la función Lambda y la invocará, ya que la pila contiene un único recurso de Lambda.
- Pila que contiene una sola cola de Amazon SQS y dos aplicaciones de Kinesis Data Streams: localizará la cola de Amazon SQS y AWS SAM CLI la invocará, ya que la pila contiene una sola cola de Amazon SQS.

Opciones

`--beta-features` | `--no-beta-features`

Permitir o denegar las características beta.

`--config-env` *TEXT*

Especifica el entorno que se va a utilizar en el archivo de configuración de la CLI de AWS SAM.

Valor predeterminado: `default`

`--config-file` *FILENAME*

Especifica la ruta y el nombre del archivo de configuración.

Para obtener más información sobre los archivos de configuración, consulta [Configuración de la AWS SAM de CLI](#).

Predeterminado: `samconfig.toml` en la raíz del directorio del proyecto.

`--debug`

Activar el registro de depuración. Esto imprime los mensajes de depuración y las marcas temporales generados por AWS SAMCLI.

`--event`, `-e` *TEXT*

El evento que se va a enviar al recurso de destino.

`--event-file` *FILENAME*

La ruta a un archivo que contiene el evento que se va a enviar al recurso de destino.

`--help`, `-h`

Muestre el mensaje de ayuda y salga.

`--output` *[text | json]*

Genera los resultados de la invocación en un formato de salida específico.

`json`: los metadatos de la solicitud y la respuesta del recurso se devuelven en una estructura JSON. La respuesta contiene el resultado completo del SDK.

`text`: los metadatos de la solicitud se devuelven en una estructura de texto. La respuesta del recurso se devuelve en el formato de salida del recurso invocado.

`--parameter`

Parámetros [Boto3](#) adicionales que puede pasar al recurso que se está invocando.

Amazon Kinesis Data Streams

Los parámetros adicionales siguientes se pueden utilizar para incluir un registro en el flujo de datos de Kinesis:

- ExplicitHashKey='string'
- PartitionKey='string'
- SequenceNumberForOrdering='string'
- StreamARN='string'

Para obtener una descripción de cada parámetro, consulte [Kinesis.Client.put_record](#).

AWS Lambda

Los siguientes parámetros adicionales se pueden usar para invocar un recurso de Lambda y recibir una respuesta almacenada en búfer:

- ClientContext='base64-encoded string'
- InvocationType='[DryRun | Event | RequestResponse]'
- LogType='[None | Tail]'
- Qualifier='string'

Los siguientes parámetros adicionales se pueden usar para invocar un recurso de Lambda con transmisión de respuesta:

- ClientContext='base64-encoded string'
- InvocationType='[DryRun | RequestResponse]'
- LogType='[None | Tail]'
- Qualifier='string'

Para ver una descripción de los parámetros, consulta lo siguiente:

- Lambda con respuesta amortiguada: [Lambda.Client.invoke](#)
- [Lambda con transmisión de respuestas: _with_response_stream](#) [Lambda.Client.invoke](#)

Amazon Simple Queue Service (Amazon SQS)

Los parámetros adicionales siguientes se pueden utilizar para enviar un mensaje a una cola de Amazon SQS:

- DelaySeconds=*integer*

- MessageAttributes='json string'
- MessageDeduplicationId='string'
- MessageGroupId='string'
- MessageSystemAttributes='json string'

[Para obtener una descripción de cada parámetro, consulte `\_message.SQS.Client.send`](#)

AWS Step Functions

Los parámetros adicionales siguientes se pueden utilizar para iniciar la ejecución de una máquina de estado:

- name='string'
- traceHeader='string'

Para obtener una descripción de cada parámetro, consulte [SFN.Client.start_execution](#).

--profile *TEXT*

El perfil específico del archivo de credenciales para obtener AWS las credenciales.

--region *TEXT*

El Región de AWS del recurso. Por ejemplo, us-east-1.

--stack-name *TEXT*

El nombre de la CloudFormation pila a la que pertenece el recurso.

--tenant-id *TEXT*

El ID de inquilino de las funciones Lambda multiusuario. Se utiliza para garantizar el aislamiento informático entre distintos inquilinos. Se requiere cuando se invocan funciones configuradas con el modo de aislamiento de inquilinos.

--test-event-name *NAME*

El nombre del evento de prueba compatible y pasarlo a la función de Lambda.

Note

Esta opción solo admite funciones de Lambda.

Ejemplo

El siguiente ejemplo invoca los recursos compatibles de la AWS nube y activa el registro de depuración, que imprime los mensajes de depuración y las marcas de tiempo generados por: AWS SAMCLI

```
$ sam remote invoke--debug
```

evento de prueba remoto de sam

Esta página proporciona información de referencia para el AWS Serverless Application Model comando Command Line Interface (AWS SAMCLI) `sam remote test-event`.

- Para obtener una introducción al AWS SAMCLI, consulte [¿Qué es el AWS SAM CLI?](#)
- Para obtener documentación sobre el uso del AWS SAMCLI `sam remote test-event` comando, consulte [Introducción a las pruebas en la nube con evento de prueba remoto de sam](#).

El `sam remote test-event` comando interactúa con los eventos de prueba que se pueden compartir en el registro de EventBridge esquemas de Amazon.

De uso

```
$ sam remote test-event <options> <subcommand>
```

Opciones

`--help, -h`

Muestre el mensaje de ayuda y salga.

Subcomandos

delete

Elimine un evento de prueba que se pueda compartir del registro del EventBridge esquema. Para obtener más información de referencia, consulta [eliminación remota del evento de prueba sam](#).

get

Obtenga un evento de prueba que se pueda compartir del registro de EventBridge esquemas. Para obtener más información de referencia, consulta [sam remote test-event get](#).

list

Enumere los eventos de prueba existentes que se pueden compartir para una función de AWS Lambda evento. Para obtener más información de referencia, consulta [lista de eventos de prueba remotos sam](#).

put

Guarde un evento de un archivo local en el registro de EventBridge esquemas. Para obtener más información de referencia, consulta [salida de evento de prueba remota sam](#).

eliminación remota del evento de prueba sam

Esta página proporciona información de referencia para el AWS Serverless Application Model `sam remote test-event delete` subcomando Command Line Interface (AWS SAMCLI).

- Para obtener una introducción al AWS SAMCLI, consulte [¿Qué es el AWS SAM CLI?](#)
- Para obtener documentación sobre el uso del AWS SAMCLI `sam remote test-event` comando, consulte [Introducción a las pruebas en la nube con evento de prueba remoto de sam](#).

El `sam remote test-event delete` subcomando elimina un evento de prueba que se puede compartir del registro de esquemas de Amazon EventBridge .

De uso

```
$ sam remote test-event delete <arguments> <options>
```

Argumentos

Resource ID (ID de recurso)

El ID de la función de AWS Lambda evento asociada al evento de prueba que se puede compartir.

Si proporciona un identificador lógico, también debe proporcionar un valor para la AWS CloudFormation pila asociada a la función Lambda mediante la `--stack-name` opción.

Valores válidos: el identificador lógico o el ARN del recurso ARN.

Opciones

`--config-env` *TEXT*

El nombre del entorno que especifica los valores de los parámetros predeterminados del archivo de configuración que se va a utilizar. El valor por defecto es «predeterminado». Para obtener más información sobre los archivos de configuración, consulta [Archivo de configuración de la CLI de AWS SAM](#).

`--config-file` *PATH*

La ruta y el nombre del archivo de configuración que contiene los valores de parámetros predeterminados que se van a utilizar. El valor por defecto es «samconfig.toml» en la raíz del directorio del proyecto. Para obtener más información sobre los archivos de configuración, consulta [Archivo de configuración de la CLI de AWS SAM](#).

`--help`, `-h`

Muestre el mensaje de ayuda y salga.

`--name` *TEXT*

El nombre del evento de prueba compatible y se va a eliminar.

`--stack-name` *TEXT*

El nombre de la CloudFormation pila asociada a la función Lambda.

Esta opción es obligatoria si proporciona el ID lógico de la función de Lambda como argumento.

sam remote test-event get

Esta página proporciona información de referencia para el AWS Serverless Application Model `sam remote test-event get` subcomando Command Line Interface (AWS SAMCLI).

- Para obtener una introducción al AWS SAMCLI, consulte [¿Qué es el AWS SAM CLI?](#)
- Para obtener documentación sobre el uso del AWS SAMCLI `sam remote test-event` comando, consulte [Introducción a las pruebas en la nube con evento de prueba remoto de sam](#).

El `sam remote test-event get` subcomando obtiene un evento de prueba que se puede compartir del registro de EventBridge esquemas de Amazon.

De uso

```
$ sam remote test-event get <arguments> <options>
```

Argumentos

Resource ID (ID de recurso)

El ID de la función de AWS Lambda evento asociada al evento de prueba que se puede compartir y que se va a obtener.

Si proporciona un identificador lógico, también debe proporcionar un valor para la AWS CloudFormation pila asociada a la función Lambda mediante la `--stack-name` opción.

Valores válidos: el identificador lógico o el ARN del recurso ARN.

Opciones

`--config-env` *TEXT*

El nombre del entorno que especifica los valores de los parámetros predeterminados del archivo de configuración que se va a utilizar. El valor por defecto es «predeterminado». Para obtener más información sobre los archivos de configuración, consulta [Archivo de configuración de la CLI de AWS SAM](#).

`--config-file` *PATH*

La ruta y el nombre del archivo de configuración que contiene los valores de parámetros predeterminados que se van a utilizar. El valor por defecto es «samconfig.toml» en la raíz del directorio del proyecto. Para obtener más información sobre los archivos de configuración, consulta [Archivo de configuración de la CLI de AWS SAM](#).

`--help`, `-h`

Muestre el mensaje de ayuda y salga.

`--name` *TEXT*

El nombre del evento de prueba compartible y que se va a obtener.

`--output-file` *FILENAME*

La ruta y el nombre del archivo en el que se guardará el evento en su máquina local.

Si no proporciona esta opción, AWS SAM CLI mostrará el contenido del evento de prueba que se puede compartir en su consola.

`--stack-name` *TEXT*

El nombre de la CloudFormation pila asociada a la función Lambda.

Esta opción es obligatoria si proporciona el ID lógico de la función de Lambda como argumento.

lista de eventos de prueba remotos sam

Esta página proporciona información de referencia para el AWS Serverless Application Model `sam remote test-event list` subcomando Command Line Interface (AWS SAMCLI).

- Para obtener una introducción al AWS SAMCLI, consulte [¿Qué es el AWS SAM CLI?](#)
- Para obtener documentación sobre el uso del AWS SAMCLI `sam remote test-event` comando, consulte [Introducción a las pruebas en la nube con evento de prueba remoto de sam](#).

El `sam remote test-event list` subcomando muestra los eventos de prueba que se pueden compartir existentes para una función de AWS Lambda eventos específica del registro de EventBridge esquemas de Amazon.

De uso

```
$ sam remote test-event list <arguments> <options>
```

Argumentos

Resource ID (ID de recurso)

El ID de la función de Lambda asociada a los eventos de prueba que se pueden compartir.

Si proporciona un identificador lógico, también debe proporcionar un valor para la AWS CloudFormation pila asociada a la función Lambda mediante la `--stack-name` opción.

Valores válidos: el identificador lógico o el ARN del recurso ARN.

Opciones

`--config-env` *TEXT*

El nombre del entorno que especifica los valores de los parámetros predeterminados del archivo de configuración que se va a utilizar. El valor por defecto es «predeterminado». Para obtener más información sobre los archivos de configuración, consulta [Archivo de configuración de la CLI de AWS SAM](#).

`--config-file` *PATH*

La ruta y el nombre del archivo de configuración que contiene los valores de parámetros predeterminados que se van a utilizar. El valor por defecto es «samconfig.toml» en la raíz del directorio del proyecto. Para obtener más información sobre los archivos de configuración, consulta [Archivo de configuración de la CLI de AWS SAM](#).

`--help`, `-h`

Muestre el mensaje de ayuda y salga.

`--stack-name` *TEXT*

El nombre de la CloudFormation pila asociada a la función Lambda.

Esta opción es obligatoria si proporciona el ID lógico de la función de Lambda como argumento.

Ejemplos

Para ver ejemplos sobre el uso de este comando, consulte [Enumeración de eventos de prueba compatibles](#).

salida de evento de prueba remota sam

Esta página proporciona información de referencia para el AWS Serverless Application Model `sam remote test-event put` subcomando Command Line Interface (AWS SAMCLI).

- Para obtener una introducción al AWS SAMCLI, consulte [¿Qué es el AWS SAM CLI?](#)
- Para obtener documentación sobre el uso del AWS SAMCLI `sam remote test-event` comando, consulte [Introducción a las pruebas en la nube con evento de prueba remoto de sam](#).

El `sam remote test-event put` subcomando guarda un evento de prueba que se puede compartir desde su máquina local en el registro de EventBridge esquemas de Amazon.

De uso

```
$ sam remote test-event put <arguments> <options>
```

Argumentos

Resource ID (ID de recurso)

El ID de la función de AWS Lambda evento asociada al evento de prueba que se puede compartir.

Si proporciona un identificador lógico, también debe proporcionar un valor para la AWS CloudFormation pila asociada a la función Lambda mediante la `--stack-name` opción.

Valores válidos: el identificador lógico o el ARN del recurso ARN.

Opciones

`--config-env` *TEXT*

El nombre del entorno que especifica los valores de los parámetros predeterminados del archivo de configuración que se va a utilizar. El valor por defecto es «predeterminado». Para obtener más información sobre los archivos de configuración, consulta [Archivo de configuración de la CLI de AWS SAM](#).

`--config-file` *PATH*

La ruta y el nombre del archivo de configuración que contiene los valores de parámetros predeterminados que se van a utilizar. El valor por defecto es «samconfig.toml» en la raíz del directorio del proyecto. Para obtener más información sobre los archivos de configuración, consulta [Archivo de configuración de la CLI de AWS SAM](#).

`--file` *FILENAME*

La ruta del archivo y el nombre del evento en su máquina local.

Proporciona - como valor del nombre del archivo a leer de stdin.

Esta opción es obligatoria.

`--force, -f`

Sobrescriba un evento de prueba compatible con el mismo nombre.

`--help, -h`

Muestre el mensaje de ayuda y salga.

`--name` *TEXT*

El nombre con el que se guardará el evento de prueba compartible.

Si existe un evento de prueba que se pueda compartir con el mismo nombre en el registro del EventBridge esquema, no AWS SAM CLI lo sobrescribirá. Para sobrescribirlo, añada la opción de `--force`.

`--output-file` *FILENAME*

La ruta y el nombre del archivo en el que se guardará el evento en su máquina local.

Si no incluyes esta opción, AWS SAM CLI se mostrará el contenido del evento de prueba que se puede compartir en tu consola.

`--stack-name` *TEXT*

El nombre de la CloudFormation pila asociada a la función Lambda.

Esta opción es obligatoria si proporciona el ID lógico de la función de Lambda como argumento.

Ejemplo

Para ver un ejemplo sobre el uso de este comando, consulte [Almacenamiento de eventos de prueba compartibles](#).

sam sync

Esta página proporciona información de referencia para el AWS Serverless Application Model comando Command Line Interface (AWS SAMCLI) `sam sync`.

- Para obtener una introducción al AWS SAMCLI, consulte [¿Qué es el AWS SAM CLI?](#)
- Para obtener documentación sobre el uso del AWS SAMCLI, consulte [AWS SAM CLI](#).

El comando `sam sync` sincroniza los cambios de la aplicación local con Nube de AWS.

De uso

```
$ sam sync <options>
```

Opciones

`--base-dir, -s` *DIRECTORY*

Resuelve las rutas relativas al código fuente de la función o capa con respecto a este directorio. Utiliza esta opción para cambiar la forma en que se resuelven las rutas relativas a las carpetas de código fuente. De forma predeterminada, las rutas relativas se resuelven con respecto a la ubicación de la AWS SAM plantilla.

Además de los recursos de la aplicación o pila raíz que esté creando, esta opción también se aplica a las aplicaciones o pilas anidadas. Además, esta opción se aplica a los siguientes tipos y propiedades de recursos:

- Tipo de recurso: `AWS::Serverless::Function` Propiedad: `CodeUri`
- Tipo de recurso: `AWS::Serverless::Function` Atributo de recurso: `Metadata Entrada: DockerContext`
- Tipo de recurso: `AWS::Serverless::LayerVersion` Propiedad: `ContentUri`
- Tipo de recurso: `AWS::Lambda::Function` Propiedad: `Code`
- Tipo de recurso: `AWS::Lambda::LayerVersion` Propiedad: `Content`

`--build-image` *TEXT*

El URI de la [imagen de contenedor](#) que quiere utilizar al crear la aplicación. De forma predeterminada, AWS SAM utiliza el URI del repositorio de imágenes del contenedor de [Amazon Elastic Container Registry \(Amazon ECR\) Public](#). Especifica esta opción para utilizar una imagen diferente.

Puede utilizar esta opción varias veces con un solo comando. Cada opción acepta una cadena o un par clave-valor.

- Cadena: especifica el URI de la imagen del contenedor que utilizarán todos los recursos de la aplicación. A continuación, se muestra un ejemplo:

```
$ sam sync --build-image amazon/aws-sam-cli-build-image-python3.8
```

- Key-value par: especifique el nombre del recurso como clave y el URI de la imagen del contenedor que se utilizará con ese recurso como valor. Utiliza este formato para especificar un URI de imagen de contenedor diferente para cada recurso de la aplicación. A continuación, se muestra un ejemplo:


```
$ sam sync --build-image Function1=amazon/aws-sam-cli-build-image-python3.8
```

Esta opción solo se aplica si se especifica la opción `--use-container`; de lo contrario, se producirá un error.

`--build-in-source` | `--no-build-in-source`

Proporciona `--build-in-source` para compilar el proyecto directamente en la carpeta de origen.

La opción `--build-in-source` admite los siguientes tiempos de ejecución y métodos de compilación:

- Tiempos de ejecución: cualquier tiempo de ejecución Node.js compatible con la opción [sam init --runtime](#).
- Métodos de compilación: Makefile, esbuild.

La opción `--build-in-source` no es compatible con las siguientes opciones:

- `--use-container`

Valor predeterminado: `--no-build-in-source`

`--capabilities` *LIST*

Una lista de capacidades que se especifican para CloudFormation permitir la creación de determinadas pilas. Algunas plantillas de pila pueden incluir recursos que pueden afectar a los permisos de tu Cuenta de AWS. Por ejemplo, mediante la creación de nuevos usuarios AWS Identity and Access Management (IAM). Especifica esta opción para anular los valores predeterminados. Entre los valores válidos se incluyen:

- `CAPABILITY_IAM`
- `CAPABILITY_NAMED_IAM`
- `CAPABILITY_RESOURCE_POLICY`
- `CAPABILITY_AUTO_EXPAND`

Valores predeterminados: `CAPABILITY_NAMED_IAM` y `CAPABILITY_AUTO_EXPAND`

`--code`

De forma predeterminada, AWS SAM sincroniza todos los recursos de la aplicación. Especifica esta opción para sincronizar solo los recursos de código, que incluyen lo siguiente:

- `AWS::Serverless::Function`

- `AWS::Lambda::Function`
- `AWS::Serverless::LayerVersion`
- `AWS::Lambda::LayerVersion`
- `AWS::Serverless::Api`
- `AWS::ApiGateway::RestApi`
- `AWS::Serverless::HttpApi`
- `AWS::ApiGatewayV2::Api`
- `AWS::Serverless::StateMachine`
- `AWS::StepFunctions::StateMachine`

Para sincronizar los recursos de código, AWS SAM utiliza las API AWS de servicio directamente, en lugar de desplegarlas de forma automática. AWS CloudFormation Para actualizar tu CloudFormation pila, ejecuta `sam sync --watch osam deploy`.

`--config-env` *TEXT*

El nombre del entorno que especifica los valores de los parámetros predeterminados del archivo de configuración que se va a utilizar. El valor por defecto es «predeterminado». Para obtener más información sobre los archivos de configuración, consulta [Archivo de configuración de la CLI de AWS SAM](#).

`--config-file` *PATH*

La ruta y el nombre del archivo de configuración que contiene los valores de parámetros predeterminados que se van a utilizar. El valor por defecto es «`samconfig.toml`» en la raíz del directorio del proyecto. Para obtener más información sobre los archivos de configuración, consulta [Archivo de configuración de la CLI de AWS SAM](#).

`--dependency-layer` | `--no-dependency-layer`

Especifica si se deben separar las dependencias de las funciones individuales en otra capa para acelerar el proceso de sincronización.

Valor predeterminado: `--dependency-layer`

`--express` | `--no-express`

Especifica si se debe usar el modo CloudFormation Express para las implementaciones de infraestructura. Cuando están activas, las operaciones de apilamiento se completan una vez que se aplica la configuración de los recursos. Los recursos siguen preparándose en segundo plano.

El modo exprés solo se aplica a la ruta de sincronización de la infraestructura. Code-only las sincronizaciones (actualizaciones de funciones de eventos de Lambda, despliegues) CloudFormation omiten esta opción y no se ven afectadas por ella.

El valor predeterminado es `--no-express`.



Próximo cambio predeterminado

El valor predeterminado cambia al 1 de `--express` septiembre de 2026.

`--image-repository` *TEXT*

El nombre del repositorio Amazon Elastic Container Registry (Amazon ECR) en el que este comando carga la imagen de la función. Es necesario para las funciones declaradas con el tipo de paquete Image.

`--image-repositories` *TEXT*

Una asignación de funciones a su URI de repositorio de Amazon ECR. Haga referencia a las funciones por su ID lógico. A continuación, se muestra un ejemplo:

```
$ sam sync --image-repositories Function1=123456789012.dkr.ecr.us-east-1.amazonaws.com/my-repo
```

Puede especificar esta opción varias veces con un solo comando.

`--kms-key-id` *TEXT*

El ID de una clave AWS Key Management Service (AWS KMS) que se utiliza para cifrar los artefactos que se encuentran en reposo en el bucket de Amazon S3. Si no especificas esta opción, utilizará las claves de S3-managed cifrado de Amazon. AWS SAM

`--metadata`

Un mapa de metadatos para adjuntar a todos los artefactos a los que haga referencia en la plantilla.

`--notification-arns` *LIST*

Una lista de los ARN de temas de Amazon Simple Notification Service (Amazon SNS) asociados a la pila CloudFormation .

`--no-use-container`

Una opción que permite usar el kit de herramientas de IDE para establecer el comportamiento predeterminado.

`--parameter-overrides`

Una cadena que contiene anulaciones de CloudFormation parámetros codificadas como pares clave-valor. Usa el mismo formato que (). AWS Command Line Interface AWS CLI El formato de la CLI de AWS SAM es de palabras clave y de valor explícitas, y cada modificación está separada por un espacio. A continuación, se incluyen dos ejemplos:

- `--parameter-overrides ParameterKey=hello,ParameterValue=world`
- `--parameter-overrides ParameterKey=hello,ParameterValue=world
ParameterKey=example1,ParameterValue=example2
ParameterKey=apple,ParameterValue=banana`

`--resource` *TEXT*

Especifica el tipo de recurso que se va a sincronizar. Para sincronizar varios recursos, puede especificar esta opción varias veces. Esta opción es compatible con la opción `--code`. El valor debe ser uno de los recursos que se enumeran en `--code`. Por ejemplo, `--resource AWS::Serverless::Function` `--resource AWS::Serverless::LayerVersion`.

`--resource-id` *TEXT*

Especifica el ID del recurso que se va a sincronizar. Para sincronizar varios recursos, puede especificar esta opción varias veces. Esta opción es compatible con la opción `--code`. Por ejemplo, `--resource-id Function1` `--resource-id Function2`.

`--role-arn` *TEXT*

El nombre de recurso de Amazon (ARN) de una función de IAM que se CloudFormation asume al aplicar el conjunto de cambios.

`--s3-bucket` *TEXT*

El nombre del depósito de Amazon Simple Storage Service (Amazon S3) en el que este comando carga CloudFormation la plantilla. Si la plantilla tiene más de 51 200 bytes, se requiere la opción `--s3-bucket` o la `--resolve-s3`. Si especificas ambas opciones `--s3-bucket` y `--resolve-s3`, se producirá un error.

`--s3-prefix` *TEXT*

El prefijo que se agrega a los nombres de los artefactos que carga en el bucket de Amazon S3. El nombre del prefijo es un nombre de ruta (nombre de carpeta) del bucket de Amazon S3. Esto se aplica únicamente a las funciones declaradas con el tipo de paquete Zip.

`--save-params`

Guarda los parámetros que proporciona en la línea de comandos en el archivo de AWS SAM configuración.

`--skip-deploy-sync` | `--no-skip-deploy-sync`

Especifica `--skip-deploy-sync` para omitir la sincronización inicial de la infraestructura si no es necesaria. AWS SAM CLI comparará su AWS SAM plantilla local con la CloudFormation plantilla desplegada y realizará una implementación solo si se detecta un cambio.

`--no-skip-deploy-sync` Especifica que se debe realizar un CloudFormation despliegue cada vez `sam sync` que se ejecute.

Para obtener más información, consulte [Omita la inicial CloudFormation Implementación](#).

Valor predeterminado: `--skip-deploy-sync`

`--stack-name` *TEXT*

El nombre de la CloudFormation pila de la aplicación.

Esta opción es obligatoria.

`--tags` *LIST*

Una lista de etiquetas para asociarlas a la pila que se crea o actualiza. CloudFormation también propaga estas etiquetas a los recursos de la pila que las admiten.

`--template-file`, `--template`, `-t` *PATH*

La ruta y el nombre del archivo donde se encuentra la AWS SAM plantilla.

 Note

Si especifica esta opción, solo se AWS SAM despliega la plantilla y los recursos locales a los que apunta.

`--use-container, -u`

Si sus funciones dependen de paquetes que tienen dependencias compiladas de forma nativa, utilice esta opción para crear la función dentro de un AWS Lambda contenedor similar. Docker

 Note

Actualmente, esta opción no es compatible con `--dependency-layer`. Si usas `--use-container` con `--dependency-layer`, la CLI de AWS SAM le informará y continuará con `--no-dependency-layer`.

`--watch`

Inicia un proceso que vigila la aplicación local en busca de cambios y los sincroniza automáticamente con la. Nube de AWS De forma predeterminada, al especificar esta opción, AWS SAM sincroniza todos los recursos de la aplicación a medida que los actualiza. Con esta opción, AWS SAM realiza un CloudFormation despliegue inicial. A continuación, AWS SAM utiliza las API AWS de servicio para actualizar los recursos de código. AWS SAM CloudFormation se utiliza para actualizar los recursos de infraestructura al actualizar la AWS SAM plantilla.

`--watch-exclude` *TEXT*

Excluye un archivo o carpeta de ser observado para cambios de archivo. Para utilizar esta opción, también se debe proporcionar `--watch`.

Esta opción recibe un par de clave-valor:

- Clave: el identificador lógico de una función de Lambda en la aplicación.
- Valor: el nombre del archivo o la carpeta asociados que se van a excluir.

Al actualizar los archivos o carpetas especificados en la `--watch-exclude` opción, no AWS SAM CLI se iniciará la sincronización. Sin embargo, cuando una actualización de otros archivos o carpetas inicie una sincronización, estos archivos o carpetas se incluirán en dicha sincronización.

Puede proporcionar esta opción varias veces en un mismo comando.

Ejemplos

Para ver ejemplos sobre el uso de este comando, consulte [Opciones para el comando de sincronización sam](#).

sam traces

Esta página proporciona información de referencia para el AWS Serverless Application Model comando Command Line Interface (AWS SAMCLI) `sam traces`.

Para obtener una introducción al AWS SAMCLI, consulte [¿Qué es el AWS SAM CLI?](#)

El `sam traces` comando busca AWS X-Ray trazas Cuenta de AWS en tu. Región de AWS

De uso

```
$ sam traces <options>
```

Opciones

`--config-env` *TEXT*

El nombre del entorno que especifica los valores de los parámetros predeterminados del archivo de configuración que se va a utilizar. El valor por defecto es «predeterminado». Para obtener más información sobre los archivos de configuración, consulta [Archivo de configuración de la CLI de AWS SAM](#).

`--config-file` *PATH*

La ruta y el nombre del archivo de configuración que contiene los valores de parámetros predeterminados que se van a utilizar. El valor por defecto es «samconfig.toml» en la raíz del directorio del proyecto. Para obtener más información sobre los archivos de configuración, consulta [Archivo de configuración de la CLI de AWS SAM](#).

`--end-time` *TEXT*

Busca rastros hasta este momento. El momento puede darse en valores relativos como «Hace 5 minutos», «mañana» o como una marca de tiempo formateada como «2018-01-01 10:10:10».

`--output` *TEXT*

Especifica el formato de salida para los registros. Para imprimir registros formateados, especifica `text`. Para imprimir los registros como JSON, especifica `json`.

`--save-params`

Guarde los parámetros que proporcione en la línea de comandos en el archivo AWS SAM de configuración.

`--start-time` *TEXT*

Busca rastros a partir de este momento. El momento puede darse en valores relativos como «Hace 5 minutos», «ayer» o como una marca de tiempo formateada como «2018-01-01 10:10:10». El valor predeterminado es «Hace 10 minutos».

`--tail`

Sigue el resultado de la traza. Esto ignora el argumento de la hora de finalización y sigue mostrando las trazas a medida que están disponibles.

`--trace-id` *TEXT*

El identificador único para un rastreo de X-Ray.

Ejemplos

Ejecuta el siguiente comando para obtener los rastros de X-Ray por ID.

```
$ sam traces --trace-id tracing-id-1 --trace-id tracing-id-2
```

Ejecuta el siguiente comando para rastrear las trazas de rayos X a medida que estén disponibles.

```
$ sam traces --tail
```

sam validate

Esta página proporciona información de referencia para el AWS Serverless Application Model comando Command Line Interface (AWS SAMCLI) `sam validate`.

Para obtener una introducción al AWS SAMCLI, consulte [¿Qué es el AWS SAM CLI?](#)

El `sam validate` comando comprueba si un archivo AWS SAM de plantilla es válido.

De uso

```
$ sam validate <options>
```


Opciones

`--config-env` *TEXT*

El nombre del entorno que especifica los valores de los parámetros predeterminados del archivo de configuración que se va a utilizar. El valor por defecto es «predeterminado». Para obtener más información sobre los archivos de configuración, consulta [Archivo de configuración de la CLI de AWS SAM](#).

`--config-file` *PATH*

La ruta y el nombre del archivo de configuración que contiene los valores de parámetros predeterminados que se van a utilizar. El valor predeterminado es «samconfig.toml» en la raíz del directorio del proyecto. Para obtener más información sobre los archivos de configuración, consulta [Archivo de configuración de la CLI de AWS SAM](#).

`--debug`

Activa el registro de depuración para imprimir el mensaje de depuración generado por la CLI de AWS SAM y muestra las marcas horarias.

`--lint`

Ejecuta la validación del elemento lint en la plantilla a través de cfn-lint. Crea un archivo de configuración `cfnlintrc` para especificar parámetros adicionales. Para obtener más información, consulte [cfn-lint en el repositorio](#). CloudFormation GitHub

`--profile` *TEXT*

El perfil específico del archivo de credenciales que recibe las credenciales. AWS

`--region` *TEXT*

La AWS región en la que se va a realizar el despliegue. Por ejemplo, us-east-1.

`--save-params`

Guarde los parámetros que proporcione en la línea de comandos en el archivo AWS SAM de configuración.

`--template-file`, `--template`, `-t` *PATH*

El archivo AWS SAM de plantilla. El valor predeterminado es `template.[yaml|yml]`.

Si la plantilla se encuentra en el directorio de trabajo actual y se denomina `template.[yaml|yml|json]`, esta opción no es necesaria.

Si acaba de ejecutar `sam build`, esta opción no es necesaria.

Ejemplo

Para ver un ejemplo sobre el uso de este comando para validar una plantilla, consulte [Validar archivos AWS SAM de plantilla](#).

Para ver un ejemplo sobre el uso de este comando con `cfn-lint`, consulte [Valide sus AWS SAM aplicaciones con CloudFormation Linter](#).

Administración de la CLI de AWS SAM

Esta sección contiene información sobre las formas en que puede administrar y personalizar su versión de AWS SAMCLI. Incluye información sobre cómo configurar los valores de los parámetros de comando de la CLI de AWS SAM mediante un archivo de configuración a nivel de proyecto. También incluye información sobre cómo administrar diferentes versiones de su AWS SAMCLI dispositivo, configurar AWS credenciales para AWS SAM poder realizar llamadas a AWS los servicios en su nombre y diferentes formas de personalizarlo AWS SAM. Esta sección termina con una sección sobre la AWS SAM solución de problemas generales.

Temas

- [Archivo de configuración de la CLI de AWS SAM](#)
- [Administración de versiones de AWS SAM CLI](#)
- [Configuración de AWS credenciales](#)
- [Telemetría en la AWS SAMCLI](#)
- [AWS SAM CLI solución de problemas](#)

Archivo de configuración de la CLI de AWS SAM

La interfaz de línea de AWS Serverless Application Model comandos (AWS SAMCLI) admite un archivo de configuración a nivel de proyecto que se puede utilizar para configurar los valores de los parámetros de los AWS SAMCLI comandos.

Para obtener documentación sobre cómo crear y usar archivos de configuración, consulta [Configuración de la AWS SAM de CLI](#).

Temas

- [Parámetros predeterminados del archivo de configuración](#)
- [Formatos de archivo de configuración admitidos](#)
- [Especificar un archivo de configuración](#)
- [Fundamentos de los archivos de configuración](#)
- [Reglas del valor del parámetro](#)
- [Prioridad de configuración](#)
- [Crear y modificar archivos de configuración](#)

Parámetros predeterminados del archivo de configuración

AWS SAM utiliza los siguientes parámetros predeterminados del archivo de configuración:

- Nombre: `samconfig`.
- Ubicación: en la raíz del proyecto. Es la misma ubicación que el archivo `template.yaml`.
- Format: TOML Para obtener más información, consulta [TOML](#) en la TOML documentación.

A continuación se muestra un ejemplo de estructura de proyecto que incluye el nombre y la ubicación por defecto del archivo de configuración:

```
sam-app
### README.md
### __init__.py
### events
### hello_world
### samconfig.toml
### template.yaml
### tests
```

A continuación se muestra un ejemplo de un archivo `samconfig.toml`:

```
...
version = 0.1

[default]
[default.global]
```

```
[default.global.parameters]
stack_name = "sam-app"

[default.build.parameters]
cached = true
parallel = true

[default.deploy.parameters]
capabilities = "CAPABILITY_IAM"
confirm_changeset = true
resolve_s3 = true

[default.sync.parameters]
watch = true

[default.local_start_api.parameters]
warm_containers = "EAGER"

[prod]
[prod.sync]
[prod.sync.parameters]
watch = false
```

Formatos de archivo de configuración admitidos

Los formatos admitidos son TOML y [YAML | YML]. Consulte la siguiente sintaxis básica:

TOML

```
version = 0.1
[environment]
[environment.command]
[environment.command.parameters]
option = parameter value
```

YAML

```
version: 0.1
environment:
  command:
    parameters:
      option: parameter value
```

Especificar un archivo de configuración

De forma predeterminada, la CLI de AWS SAM busca un archivo de configuración en el siguiente orden:

1. Archivo de configuración personalizado: si utiliza la opción `--config-file` para especificar un nombre y una ubicación de archivo, la CLI de AWS SAM busca primero este archivo.
2. Archivo predeterminado **`samconfig.toml`**: este es el nombre y el formato del archivo de configuración predeterminados y se encuentra en la raíz del proyecto. Si no especifica un archivo de configuración personalizado, la CLI de AWS SAM buscará este archivo a continuación.
3. Archivo **`samconfig.[yaml|yml]`**: si `samconfig.toml` no existe en la raíz del proyecto, AWS SAMCLI busca este archivo.

El siguiente es un ejemplo de cómo se especifica un archivo de configuración personalizado mediante la opción `--config-file`:

```
$ sam deploy --config-file myconfig.yaml
```

Note

El `--config-file` parámetro debe ser relativo a la ubicación del archivo de AWS SAM plantilla porque AWS SAMCLI necesita determinar el contexto en el que se aplica la configuración. El `samconfig.toml` archivo administra los ajustes de configuración de su versión y la CLI busca el `samconfig.toml` archivo (o el parámetro del archivo de configuración anulado) en la carpeta relativa del `template.yaml` archivo. AWS SAMCLI

Fundamentos de los archivos de configuración

Entorno

Un entorno es un identificador con nombre que contiene un conjunto único de valores de configuración. Puede tener varios entornos en una sola AWS SAM aplicación.

El valor predeterminado del nombre del entorno es `default`.

Utilice la AWS SAMCLI `--config-env` opción para especificar el entorno que se va a utilizar.

Comando

El comando es el comando de la CLI de AWS SAM para el que se especifican los valores de los parámetros.

Para especificar los valores de los parámetros de todos los comandos, utilice el identificador `global`.

Al hacer referencia a un comando AWS SAMCLI, sustituya los espacios () y los guiones (–) por guiones bajos (_). Consulte los siguientes ejemplos:

- `build`
- `local_invoke`
- `local_start_api`

Parameters

Los parámetros se especifican como pares de clave-valor.

- La clave es el nombre de la opción de comando AWS SAMCLI.
- El valor es el valor que se debe especificar.

Al especificar las claves, utilice el nombre de la opción de comando de formato largo y sustituya los guiones (–) por guiones bajos (_). A continuación se muestran algunos ejemplos:

- `region`
- `stack_name`
- `template_file`

Reglas del valor del parámetro

TOML

- Los valores booleanos pueden ser `true` o `false`. Por ejemplo, `confirm_changeset = true`.
- Para los valores de cadena, utilice comillas ("). Por ejemplo, `region = "us-west-2"`.
- Para los valores de lista, utilice comillas (") y separe cada valor con un espacio (). Por ejemplo: `capabilities = "CAPABILITY_IAM CAPABILITY_NAMED_IAM"`.

- En el caso de los valores que contienen una lista de pares clave-valor, los pares están delimitados por espacios () y el valor de cada par va entre comillas codificadas (`\\" \"`). Por ejemplo, `tags = "project=\"my-application\" stage=\"production\""`.
- En el caso de los valores de los parámetros que se pueden especificar varias veces, el valor es una matriz de argumentos. Por ejemplo: `image_repositories = ["my-function-1=image-repo-1", "my-function-2=image-repo-2"]`.

YAML

- Los valores booleanos pueden ser `true` o `false`. Por ejemplo, `confirm_changeset: true`.
- Para las entradas que contienen un único valor de cadena, las comillas (") son opcionales. Por ejemplo, `region: us-west-2`. Esto incluye las entradas que contienen varios pares clave-valor que se proporcionan como una sola cadena. A continuación, se muestra un ejemplo:

```
$ sam deploy --tags "foo=bar hello=world"
```

```
default:
  deploy:
    parameters:
      tags: foo=bar hello=world
```

- Para las entradas que contienen una lista de valores o las entradas que se pueden usar varias veces en un solo comando, especifíquelas como una lista de cadenas.

A continuación, se muestra un ejemplo:

```
$ sam remote invoke --parameter "InvocationType=Event" --parameter "LogType=None"
```

```
default:
  remote_invoke:
    parameters:
      parameter:
        - InvocationType=Event
        - LogType=None
```

Prioridad de configuración

Al configurar los valores, se produce la siguiente prioridad:

- Los valores de los parámetros que se proporcionan en la línea de comandos tienen prioridad sobre los valores correspondientes del archivo de configuración y de la sección `Parameters` del archivo de plantilla.
- Si la opción `--parameter-overrides` se usa en la línea de comandos o en el archivo de configuración con la clave `parameter_overrides`, sus valores tienen prioridad sobre los valores de la sección `Parameters` del archivo de plantilla.
- En el archivo de configuración, las entradas proporcionadas para un comando específico tienen prioridad sobre las entradas globales. En el siguiente ejemplo, el comando `sam deploy` utilizará el nombre de la pila `my-app-stack`.

TOML

```
[default.global.parameters]
stack_name = "common-stack"

[default.deploy.parameters]
stack_name = "my-app-stack"
```

YAML

```
default:
  global:
    parameters:
      stack_name: common-stack
  deploy:
    parameters:
      stack_name: my-app-stack
```

Crear y modificar archivos de configuración

Creación de archivos de configuración

Al crear una aplicación utilizando `sam init`, se crea un archivo `samconfig.toml` predeterminado. También puede crear su archivo de configuración manualmente.

Modificación de los archivos de configuración

También puede modificar su archivo de configuración manualmente. Además, durante cualquier flujo interactivo AWS SAMCLI, los valores configurados se mostrarán entre corchetes ([]). Si modifica estos valores, la CLI de AWS SAM actualizará el archivo de configuración.

A continuación, se muestra un ejemplo de flujo interactivo con el comando `sam deploy --guided`.

```
$ sam deploy --guided
```

```
Configuring SAM deploy
```

```
=====
```

```
Looking for config file [samconfig.toml] : Found
```

```
Reading default arguments : Success
```

```
Setting default arguments for 'sam deploy'
```

```
=====
```

```
Stack Name [sam-app]: ENTER
```

```
AWS Region [us-west-2]: ENTER
```

```
#Shows you resources changes to be deployed and require a 'Y' to initiate deploy
```

```
Confirm changes before deploy [Y/n]: n
```

```
#SAM needs permission to be able to create roles to connect to the resources in  
your template
```

```
Allow SAM CLI IAM role creation [Y/n]: ENTER
```

```
#Preserves the state of previously provisioned resources when an operation fails
```

```
Disable rollback [y/N]: ENTER
```

```
HelloWorldFunction may not have authorization defined, Is this okay? [y/N]: y
```

```
Save arguments to configuration file [Y/n]: ENTER
```

```
SAM configuration file [samconfig.toml]: ENTER
```

```
SAM configuration environment [default]: ENTER
```

Al modificar el archivo de configuración, la CLI de AWS SAM gestiona los valores globales de la siguiente manera:

- Si el valor del parámetro existe en la sección `global` del archivo de configuración, la CLI de AWS SAM no escribe el valor en la sección de comandos específica.
- Si el valor del parámetro existe tanto en la sección de comandos específica como en la sección `global`, la CLI de AWS SAM elimina la entrada específica en favor del valor global.

Administración de versiones de AWS SAM CLI

Administre las versiones de la interfaz de línea de AWS Serverless Application Model comandos (AWS SAMCLI) mediante la actualización, la degradación y la desinstalación. Si lo deseas, puedes descargar e instalar la versión nocturna de AWS SAM CLI.

Temas

- [Actualización de AWS SAM CLI](#)
- [Desinstalación de la AWS SAM CLI](#)
- [Cambia de usar Homebrew para administrar la AWS SAM CLI](#)
- [Administrar la compilación nocturna de la CLI de AWS SAM](#)
- [Instalación de la CLI de AWS SAM en un entorno virtual mediante pip](#)
- [Administración de CLI de AWS SAM con la Homebrew](#)
- [Resolución de problemas](#)

Actualización de AWS SAM CLI

Linux

Para actualizar la AWS SAM CLI en Linux, sigue las instrucciones de instalación en [Instalación de la AWS SAM CLI](#), pero añade la opción `--update` al comando `install` de la siguiente manera:

```
sudo ./sam-installation/install --update
```

macOS

AWS SAMCLIDebe actualizarse mediante el mismo método utilizado para instalarlo. Se recomienda utilizar el instalador del paquete para instalar y actualizar el AWS SAMCLI.

Para actualizar la AWS SAM CLI mediante el instalador de paquetes, instala la última versión del paquete. Para obtener instrucciones, consulta [Instalación de la AWS SAM CLI](#).

Windows

Para actualizarla AWS SAMCLI, repita de [Instala la AWS SAM CLI](#) nuevo los pasos de instalación de Windows.

Desinstalación de la AWS SAM CLI

Linux

Para desinstalar la AWS SAM CLI en Linux, debes eliminar el enlace simbólico y el directorio de instalación ejecutando los siguientes comandos:

1. Busca el enlace simbólico y las rutas de instalación.

- Utilice el comando `which` para encontrar el enlace simbólico:

```
which sam
```

El resultado muestra la ruta en la que se encuentran AWS SAM los binarios, por ejemplo:

```
/usr/local/bin/sam
```

- Busca el directorio al que apunta el enlace simbólico mediante el comando `ls`:

```
ls -l /usr/local/bin/sam
```

En el siguiente ejemplo, el directorio de instalación es `/usr/local/aws-sam-cli`.

```
lrwxrwxrwx 1 ec2-user ec2-user 49 Oct 22 09:49 /usr/local/bin/sam -> /usr/local/  
aws-sam-cli/current/bin/sam
```

2. Elimina el enlace simbólico.

```
sudo rm /usr/local/bin/sam
```

3. Elimina el directorio de instalación.

```
sudo rm -rf /usr/local/aws-sam-cli
```

macOS

Desinstala la AWS SAM CLI con el mismo método que utilizó para instalarla. Le recomendamos que utilice el instalador del paquete para instalar el AWS SAMCLI.

Si instalaste la AWS SAM CLI mediante el instalador del paquete, sigue estos pasos para desinstalarlo.

Para desinstalar la AWS SAM CLI

1. Elimina la AWS SAM CLI modificando y ejecutando lo siguiente:

```
$ sudo rm -rf /path-to/aws-sam-cli
```

- a. **sudo**— Si su usuario tiene permisos de escritura en el lugar donde está instalado el AWS SAMCLI programa, no sudo es obligatorio. De lo contrario, se requiere el sudo.
 - b. **/path-to**— Ruta al lugar donde instaló el AWS SAMCLI programa. La ubicación predeterminada es `/usr/local`.
2. Elimine el AWS SAMCLI \$PATH modificando y ejecutando lo siguiente:

```
$ sudo rm -rf /path-to-symlink-directory/sam
```

- a. **sudo**— Si su usuario tiene permisos de escritura para \$PATH, no sudo es obligatorio. De lo contrario, se requiere el sudo.
 - b. **path-to-symlink-directory**— Tu variable de \$PATH entorno. La ubicación predeterminada es `/usr/local/bin`.
3. Comprueba que la AWS SAM CLI está desinstalada ejecutando lo siguiente:

```
$ sam --version  
command not found: sam
```

Windows

Para desinstalar la AWS SAM CLI mediante la configuración de Windows, sigue estos pasos:

1. En el menú Inicio, busque “Agregar o quitar programas”.
2. Elige el resultado denominado AWS SAM Interfaz de línea de comandos de y, a continuación, selecciona Desinstalar para iniciar el desinstalador.
3. Confirme que desea desinstalar el AWS SAMCLI.

Cambia de usar Homebrew para administrar la AWS SAM CLI

Si la utiliza Homebrew para instalar y actualizar AWS SAMCLI, le recomendamos que utilice un método AWS compatible. Siga estas instrucciones para cambiar a un método compatible.

Para dejar de usar Homebrew

1. Siga las instrucciones que aparecen en [Desinstalar una CLI Homebrew instalada AWS SAM](#) para desinstalar la versión Homebrew gestionada.
2. Siga las instrucciones que aparecen en [Instala la AWS SAM CLI](#) para instalar la CLI de AWS SAM mediante un método compatible.

Administrar la compilación nocturna de la CLI de AWS SAM

Puedes descargar e instalar la compilación nocturna de la CLI de AWS SAM. Contienes una versión preliminar del código de la CLI de AWS SAM que puedes ser menos estable que la versión de producción. Una vez instalada, puedes usar la compilación nocturna con el comando `sam-nightly`. Puedes instalar y usar la versión de producción y la versión de compilación nocturna de la CLI de AWS SAM al mismo tiempo.

Note

La compilación nocturna no contiene una versión preliminar de la imagen de compilación. Por eso, al crear una aplicación sin servidor con la opción `--use-container`, se utiliza la última versión de producción de la imagen de compilación.

Instalación de la compilación nocturna de la CLI de AWS SAM

Para instalar la compilación nocturna de la CLI de AWS SAM, sigue estas instrucciones:

Linux

Puedes instalar la versión de compilación nocturna de la CLI de AWS SAM en la plataforma Linux x86_64 mediante el instalador de paquetes.

Cómo instalar la compilación nocturna de AWS SAM CLI

1. Descargue el instalador del paquete desde [sam-cli-nightly](#) el aws-sam-cli GitHub repositorio.
2. [Sigue los pasos para instalar el paquete de compilación nocturna de la CLI de AWS SAM.](#)

macOS

Puedes instalar la versión de compilación nocturna de la CLI de AWS SAM en macOS mediante el instalador del paquete de compilación nocturna.

Cómo instalar la compilación nocturna de la AWS SAM CLI

1. Descargue el instalador del paquete para su plataforma desde [sam-cli-nightly](#) el aws-sam-cli GitHub repositorio.
2. [Sigue los pasos para instalar el paquete de compilación nocturna de la CLI de AWS SAM.](#)

Windows

La versión de compilación nocturna de la CLI de AWS SAM está disponible en este enlace de descarga: [compilación nocturna de la CLI de AWS SAM](#). Para instalar la compilación nocturna en Windows, sigue los mismos pasos que en [Instala la AWS SAM CLI](#), pero utiliza en su lugar el enlace de descarga de la compilación nocturna.

Para comprobar que has instalado la versión de compilación nocturna, ejecuta el comando `sam-nightly --version`. El resultado de este comando tienes el formato `1.X.Y.dev<YYYYMMDDHHmm>`, por ejemplo:

```
SAM CLI, version 1.20.0.dev202103151200
```

Cambia del Homebrew al instalador del paquete

Si vas a instalar Homebrew y actualizar la compilación nocturna de la CLI de AWS SAM y deseas pasar a usar el instalador de paquetes, sigue estos pasos.

Para cambiar del Homebrew al instalador de paquetes

1. Desinstala la Homebrew instalada de la CLI de AWS SAM.

```
$ brew uninstall aws-sam-cli-nightly
```

2. Comprueba que la compilación nocturna de la CLI de AWS SAM esté desinstalada ejecutando lo siguiente:

```
$ sam-nightly --version
zsh: command not found: sam-nightly
```

3. Siga los pasos de la sección anterior para instalar la compilación nocturna CLI de AWS SAM.

Instalación de la CLI de AWS SAM en un entorno virtual mediante pip

Recomendamos utilizar el instalador de paquetes nativo para instalar el AWS SAMCLI. Si debes utilizar pip, le recomendamos que instale la CLI de AWS SAM en un entorno virtual. Esto garantiza un entorno de instalación limpio y un entorno aislado en caso de que se produzcan errores.

Note

A partir del 24 de octubre de 2023, AWS SAM CLI dejará de ofrecer soporte para Python 3.7. Para obtener más información, consulta [La CLI de AWS SAM deja de dar soporte para Python 3.7.](#)

Para instalar la CLI de AWS SAM en un entorno virtual

1. Desde el directorio de inicio que eliges, crea un entorno virtual y asígnale un nombre.

Linux / macOS

```
$ mkdir project
$ cd project
$ python3 -m venv venv
```

Windows

```
> mkdir project
> cd project
> py -3 -m venv venv
```

2. Activa el entorno virtual.

Linux / macOS

```
$ . venv/bin/activate
```

El símbolo del sistema cambia para mostrar que el entorno virtual está activo.

```
(venv) $
```

Windows

```
> venv\Scripts\activate
```

El símbolo del sistema cambia para mostrar que el entorno virtual está activo.

```
(venv) >
```

3. Instala la CLI de AWS SAM en el entorno virtual.

```
(venv) $ pip install --upgrade aws-sam-cli
```

4. Comprueba que la CLI de AWS SAM esté instalada correctamente.

```
(venv) $ sam --version  
SAM CLI, version 1.94.0
```

5. Puedes utilizar el comando `deactivate` para salir del entorno virtual. Siempre que inicies una sesión nueva, debes activar el entorno de nuevo.

Administración de CLI de AWS SAM con la Homebrew

Note

A partir de septiembre de 2023, AWS dejará de mantener el Homebrew instalador AWS gestionado de AWS SAMCLI (`aws/tap/aws-sam-cli`). Para seguir utilizándolo Homebrew, puedes utilizar el instalador gestionado por la comunidad (`aws-sam-cli`). A partir de septiembre de 2023, cualquier comando Homebrew al que haga referencia a `aws-sam-cli` se redirigirá a `aws/tap/aws-sam-cli`. Te recomendamos que utilices nuestros métodos de [instalación](#) y [actualización](#) compatibles.

Instalación de la CLI de AWS SAM mediante Homebrew

Note

Estas instrucciones utilizan el AWS SAMCLI Homebrew instalador administrado por la comunidad. Para obtener más asistencia, consulta el repositorio [homebrew-core](#).

Para instalar la CLI de AWS SAM

1. Ejecuta lo siguiente:

```
$ brew install aws-sam-cli
```

2. Verificar la instalación:

```
$ sam --version
```

Tras una instalación correcta del AWS SAMCLI, debería ver un resultado como el siguiente:

```
SAM CLI, version 1.94.0
```

Actualización de la CLI de AWS SAM utilizando Homebrew

Para actualizar la CLI de AWS SAM con Homebrew, ejecuta el siguiente comando:

```
$ brew upgrade aws-sam-cli
```

Desinstalar una CLI Homebrew instalada AWS SAM

Si la CLI de AWS SAM se instaló utilizando Homebrew, sigue estos pasos para desinstalarla.

Para desinstalar la AWS SAM CLI

1. Ejecuta lo siguiente:

```
$ brew uninstall aws-sam-cli
```

2. Comprueba que la AWS SAM CLI está desinstalada ejecutando lo siguiente:

```
$ sam --version  
command not found: sam
```

Cambiar al instalador Homebrew gestionado por la comunidad

Si utiliza el Homebrew instalador AWS gestionado (`aws/tap/aws-sam-cli`) y prefiere seguir utilizándolo Homebrew, le recomendamos que cambie al Homebrew instalador gestionado por la comunidad (`aws-sam-cli`).

Para cambiar a un único comando, ejecuta lo siguiente:

```
$ brew uninstall aws-sam-cli && brew untap aws/tap && brew cleanup aws/tap && brew  
update && brew install aws-sam-cli
```

Sigue estas instrucciones para ejecutar cada comando de forma individual.

Para cambiar al instalador Homebrew gestionado por la comunidad

1. Desinstale la Homebrew versión AWS gestionada de AWS SAMCLI:

```
$ brew uninstall aws-sam-cli
```

2. Comprueba que la CLI de AWS SAM se haya desinstalado:

```
$ which sam  
sam not found
```

3. Quite el AWS SAMCLI grifo AWS gestionado:

```
$ brew untap aws/tap
```

Si recibe un error como el siguiente, añada la opción `--force` e inténtelo de nuevo.

```
Error: Refusing to untap aws/tap because it contains the following installed  
formulae or casks:  
aws-sam-cli-nightly
```

4. Elimine los archivos en caché del instalador AWS gestionado:

```
$ brew cleanup aws/tap
```

5. Actualización de Homebrew y todas las fórmulas:

```
$ brew update
```

6. Instale la versión gestionada por la comunidad de AWS SAMCLI:

```
$ brew install aws-sam-cli
```

7. Comprueba que la CLI de AWS SAM se ha realizado correctamente:

```
$ sam --version  
SAM CLI, version 1.94.0
```

Resolución de problemas

Si encuentra errores al instalar o utilizar la AWS SAMCLI, consulte [AWS SAM CLI solución de problemas](#).

Configuración de AWS credenciales

La interfaz de línea de AWS SAM comandos (CLI) requiere que establezca AWS credenciales para poder realizar llamadas a AWS los servicios en su nombre. Por ejemplo, AWS SAMCLI hace llamadas a Amazon S3 y CloudFormation.

Es posible que ya haya establecido AWS las credenciales para trabajar con AWS herramientas, como una de las AWS SDKs o las AWS CLI. Si no lo ha hecho, en este tema se muestran los enfoques recomendados para configurar AWS las credenciales.

Para establecer AWS las credenciales, debe tener el ID de la clave de acceso y la clave de acceso secreta del usuario de IAM que desee configurar. Para obtener información sobre la clave de acceso IDs y las claves de acceso secretas, consulte [Gestión de las claves de acceso para los usuarios de IAM](#) en la Guía del usuario de IAM.

A continuación, determine si la AWS CLI tiene instalada. A continuación, sigue las instrucciones en una de las siguientes secciones:

Usando el AWS CLI

Si la tiene AWS CLI instalada, utilice el `aws configure` comando y siga las instrucciones:

```
$ aws configure
AWS Access Key ID [None]: your_access_key_id
AWS Secret Access Key [None]: your_secret_access_key
Default region name [None]:
Default output format [None]:
```

Para obtener más información sobre el comando de `aws configure`, consulta [Configuración rápida de la AWS CLI](#) en la Guía del usuario de AWS Command Line Interface .

¿No está utilizando el AWS CLI

Si no la tiene AWS CLI instalada, puede crear un archivo de credenciales o establecer variables de entorno:

- Archivo de credenciales: puede establecer las credenciales en el archivo de AWS credenciales de su sistema local. Este archivo debe estar ubicado en una de las siguientes ubicaciones:
 - `~/.aws/credentials` en Linux o macOS
 - `C:\Users\USERNAME\.aws\credentials` en Windows

Este archivo debe contener líneas con el siguiente formato:

```
[default]
aws_access_key_id = your_access_key_id
aws_secret_access_key = your_secret_access_key
```

- Variables de entorno: puedes configurar las variables de entorno `AWS_ACCESS_KEY_ID` y `AWS_SECRET_ACCESS_KEY`.

Para configurar estas variables en Linux o macOS, utiliza el comando `export`:

```
export AWS_ACCESS_KEY_ID=your_access_key_id
export AWS_SECRET_ACCESS_KEY=your_secret_access_key
```

Para establecer estas variables en Windows, usa el comando `set`:

```
set AWS_ACCESS_KEY_ID=your_access_key_id  
set AWS_SECRET_ACCESS_KEY=your_secret_access_key
```

Telemetría en la AWS SAMCLI

En AWS, desarrollamos y lanzamos servicios en función de lo que aprendemos de las interacciones con los clientes. Utilizamos los comentarios de los clientes para iterar nuestro producto. La telemetría es información adicional que ayuda a comprender mejor los requisitos del cliente, diagnosticar problemas y ofrecer características para mejorar la experiencia del cliente.

La interfaz de línea de AWS SAM comandos (CLI) recopila la telemetría, como las métricas de uso genéricas, la información del sistema y el entorno y los errores. Para obtener más información sobre los tipos de telemetría recopilados, consulta [Tipos de información recopilada](#).

La CLI de AWS SAM no recopila información personal, como nombres de usuario o direcciones de correo electrónico. Tampoco extrae información confidencial en el nivel de proyecto.

Como cliente, puede controlar si la telemetría está activada y puede cambiar la configuración en cualquier momento. Si la telemetría permanece activada, AWS SAM CLI envía los datos de telemetría en segundo plano sin requerir ninguna interacción adicional con el cliente.

Desactivar la telemetría para una sola sesión

En los sistemas operativos macOS y Linux, puede desactivar la telemetría para una sola sesión. Para desactivar la telemetría de la sesión actual, ejecuta el siguiente comando para establecer la variable de entorno SAM_CLI_TELEMETRY en `false`. Repita el comando para cada nuevo terminal o sesión.

```
export SAM_CLI_TELEMETRY=0
```

Desactivar la telemetría del perfil en todas las sesiones

Ejecuta los siguientes comandos para desactivar la telemetría en todas las sesiones cuando ejecuta AWS SAM CLI en el sistema operativo.

Para desactivar la telemetría en Linux

1. Ejecuta:

```
echo "export SAM_CLI_TELEMETRY=0" >> ~/.profile
```

2. Ejecuta:

```
source ~/.profile
```

Para desactivar la telemetría en macOS

1. Ejecuta:

```
echo "export SAM_CLI_TELEMETRY=0" >> ~/.profile
```

2. Ejecuta:

```
source ~/.profile
```

Para desactivar la telemetría en Windows

Puede configurar la variable de entorno temporalmente durante la vida útil de la ventana de terminal con el siguiente comando:

Si utiliza la línea de comandos:

```
set SAM_CLI_TELEMETRY=0
```

Si se utiliza: PowerShell

```
$env:SAM_CLI_TELEMETRY=0
```

Para configurar la variable de entorno de forma permanente en la línea de comandos o PowerShell, utilice el siguiente comando:

```
setx SAM_CLI_TELEMETRY 0
```

Note

Los cambios no entrarán en vigor hasta que se cierre y se vuelva a abrir la terminal.

Tipos de información recopilada

- Información de uso: los comandos y subcomandos genéricos que ejecutan los clientes.
- Errores e información de diagnóstico: el estado y la duración de los comandos que ejecutan los clientes, incluidos los códigos de salida, los nombres de las excepciones internas y los errores al conectarse a Docker.
- Información del sistema y del entorno: la versión de Python, el sistema operativo (Windows, Linux o macOS), el entorno en el que AWS SAMCLI se ejecuta (por ejemplo AWS CodeBuild, un kit de herramientas AWS IDE o un terminal) y los valores hash de los atributos de uso.

Más información

Los datos de telemetría que AWS SAMCLI recopila cumplen con las políticas de privacidad de datos. AWS Para obtener más información, consulte los siguientes temas:

- [AWS Términos del servicio](#)
- [Preguntas frecuentes sobre privacidad de datos](#)

AWS SAM CLI solución de problemas

Esta sección proporciona detalles sobre cómo solucionar los mensajes de error al usar, instalar y administrar la interfaz de línea de AWS Serverless Application Model comandos (AWS SAMCLI).

Temas

- [Resolución de problemas](#)
- [Mensajes de error](#)
- [Mensajes de advertencia](#)

Resolución de problemas

Para obtener instrucciones de solución de problemas relacionadas con AWS SAMCLI, consulte [Solución de errores de la instalación](#).

Mensajes de error

Error de Curl: «curl: (6) No se pudo resolver:...»

Al intentar invocar el punto de conexión de API Gateway, verá el siguiente error:

```
curl: (6) Could not resolve: endpointdomain (Domain name not found)
```

Esto significa que ha intentado enviar una solicitud a un dominio que no es válido. Esto puede ocurrir si la aplicación sin servidor no se pudo implementar correctamente o si el comando curl cometió un error tipográfico. Compruebe que la aplicación se ha desplegado correctamente mediante la CloudFormation consola o la AWS CLI, y compruebe que curl el comando es correcto.

Error: no se puede encontrar la información exacta del recurso con el nombre de pila dado

Al ejecutar el `sam remote invoke` comando en una aplicación que contiene un único recurso de función de eventos de Lambda, aparece el siguiente error:

```
Error: Can't find exact resource information with given <stack-name>. Please provide full resource ARN or --stack-name to resolve the ambiguity.
```

Causa posible: no proporcionó la opción **--stack-name**.

Si el ARN de una función no se proporciona como argumento, el comando `sam remote invoke` requiere que se proporciona la opción `--stack-name`.

Solución: proporciona la opción **--stack-name**.

A continuación, se muestra un ejemplo:

```
$ sam remote invoke --stack-name sam-app
```

```
Invoking Lambda Function HelloWorldFunction
```

```
START RequestId: 40593abb-e1ad-4d99-87bd-ac032e364e82 Version: $LATEST  
END RequestId: 40593abb-e1ad-4d99-87bd-ac032e364e82  
REPORT RequestId: 40593abb-e1ad-4d99-87bd-ac032e364e82 Duration: 11.31 ms  
Billed Duration: 12 ms Memory Size: 128 MB Max Memory Used: 67 MB Init  
Duration: 171.71 ms
```



```
{"statusCode":200,"body":{"message\\":\\"hello world\\"}}}%
```

Error: no se puede encontrar la información exacta del recurso con ese nombre de pila

Al ejecutar el comando `sam remote invoke` y pasar el ARN de una función de Lambda como argumento, aparece el siguiente error:

```
Error: Can't find resource information from stack name (<stack-name>) and resource id (<function-id>)
```

Causa posible: el valor del nombre de la pila está definido en el archivo **samconfig.toml**.

La primera CLI de AWS SAM comprueba el nombre de la pila en el archivo `samconfig.toml`. Si se especifica, el argumento se pasa como un valor de ID lógico.

Solución: en su lugar, pasa el ID lógico de la función.

Puede pasar el ID lógico de la función como argumento en lugar del ARN de la función.

Solución: elimine el valor del nombre de la pila del archivo de configuración.

Puede eliminar el valor del nombre de la pila del archivo de configuración. Esto evita que AWS SAMCLI pase el ARN de la función como un valor de ID lógico.

Ejecute `sam build` después de modificar el archivo de configuración.

Error: «Error al crear recursos administrados: no se pueden localizar las credenciales»

Al ejecutar el comando `sam deploy`, aparece el siguiente error:

```
Error: Failed to create managed resources: Unable to locate credentials
```

Esto significa que no ha configurado AWS las credenciales que le permitan AWS SAMCLI realizar llamadas AWS de servicio. Para solucionar este problema, debe configurar AWS las credenciales. Para obtener más información, consulte [Configuración de AWS credenciales](#).

Error: `FileNotFoundException` en Windows

Al ejecutar comandos con la CLI de AWS SAM en Windows, podría ver el siguiente error:

```
Error: FileNotFoundException
```

Causa posible: AWS SAMCLI podría interactuar con las rutas de archivo que superen el límite máximo de rutas de acceso de Windows.

Solución: para resolver este problema, debe configurar el comportamiento de las nuevas rutas de acceso largo. Para ello, consulte [Habilitación de rutas de acceso largas en Windows 10, versión 1607 y versiones posteriores](#) en la documentación de Desarrollo de aplicaciones para Microsoft Windows.

Error: el solucionador de dependencias de pip...

Ejemplo de texto de error:

```
ERROR: pip's dependency resolver does not currently take into account all the packages
that are installed. This behaviour is the source of the following dependency
conflicts.
aws-sam-cli 1.58.0 requires aws-sam-translator==1.51.0, but you have aws-sam-translator
1.58.0 which is incompatible.
aws-sam-cli 1.58.0 requires typing-extensions==3.10.0.0, but you have typing-extensions
4.4.0 which is incompatible.
```

Causa posible: si se utiliza pip para instalar paquetes, las dependencias entre los paquetes pueden entrar en conflicto.

Cada versión del paquete `aws-sam-cli` depende de una versión del paquete `aws-sam-translator`. Por ejemplo, la versión `aws-sam-cli 1.58.0` puede depender de la versión `1.51.0` `aws-sam-translator`.

Si instala el paquete AWS SAMCLI usando pip y, a continuación, instala otro paquete que dependa de una versión más reciente de `aws-sam-translator`, ocurrirá lo siguiente:

- Se instalará la versión más reciente de `aws-sam-translator`.
- Es posible que la versión actual de `aws-sam-cli` y la versión más reciente de `aws-sam-translator` no sean compatibles.
- Cuando utilice la AWS SAMCLI, se producirá el error del solucionador de dependencias.

Soluciones:

1. Utiliza el instalador de paquetes nativo AWS SAMCLI.
 - a. Desinstalación de la AWS SAM CLI con pip. Para obtener instrucciones, consulta [Desinstalación de la AWS SAM CLI](#).

- b. Instale AWS SAMCLI mediante el instalador de paquetes nativo. Para obtener instrucciones, consulta [Instala la AWS SAM CLI](#).
 - c. Cuando sea necesario, AWS SAMCLI actualice con el instalador de paquetes nativo. Para obtener instrucciones, consulta [Actualización de AWS SAM CLI](#).
2. Si debe utilizar pip, te recomendamos que instale la CLI AWS SAM en un entorno virtual. Esto garantiza un entorno de instalación limpio y un entorno aislado en caso de que se produzcan errores. Para obtener instrucciones, consulta [Instalación de la CLI de AWS SAM en un entorno virtual mediante pip](#).

Error: No existe ese comando «remote»

Al ejecutar el comando `sam remote invoke`, aparece el siguiente error:

```
$ sam remote invoke ...
2023-06-20 08:15:07 Command remote not available
Usage: sam [OPTIONS] COMMAND [ARGS]...
Try 'sam -h' for help.

Error: No such command 'remote'.
```

Causa posible: su versión de AWS SAMCLI no está actualizada.

El AWS SAMCLI `sam remote invoke` comando se publicó con la AWS SAMCLI versión 1.88.0. Puede comprobar la versión con el comando `sam --version`.

Solución: actualice su AWS SAMCLI a la versión más reciente.

Para obtener instrucciones, consulte [Actualización de AWS SAM CLI](#).

Error: en ejecución AWS Los proyectos SAM requieren localmente Docker. ¿Lo tienes instalado?

Al ejecutar el comando `sam local start-api`, aparece el siguiente error:

```
Error: Running AWS SAM projects locally requires Docker. Have you got it installed?
```

Esto significa que no ha instalado Docker correctamente. Docker es necesario para probar la aplicación localmente. Para solucionar este problema, debe seguir las instrucciones de instalación de Docker en el host de desarrollo. Para obtener más información, consulta [Instalar Docker](#).

Error: no se cumplen las restricciones de seguridad

Al ejecutar `sam deploy --guided`, se le preguntará *Function* may not have authorization defined, Is this okay? [y/N]. Si responde a esta solicitud con **N** (la respuesta predeterminada), verá el siguiente error:

```
Error: Security Constraints Not Satisfied
```

El mensaje le informa de que la aplicación que va a implementar podría tener una API de Amazon API Gateway de acceso público configurada sin autorización. Al responder **N** a esta pregunta, está diciendo que esto no está bien.

Para resolver este problema, dispones de las siguientes opciones:

- Configura tu aplicación con autorización. Para obtener más información acerca de la autorización, consulta [Controle el acceso a la API con su AWS SAM plantilla](#).
- Si su intención es tener un punto de conexión de API de acceso público sin autorización, reinicie la implementación y responda a esta pregunta indicando que está de acuerdo con la implementación.
Y

mensaje: Falta el token de autenticación

Al intentar invocar el punto de conexión de API Gateway, verá el siguiente error:

```
{"message": "Missing Authentication Token"}
```

Esto significa que ha intentado enviar una solicitud al dominio correcto, pero el URI no es reconocible. Para solucionar este problema, verifique la URL completa y actualice el comando curl con la URL correcta.

Mensajes de advertencia

Advertencia:... AWS ya no mantendrá el Homebrew instalador para AWS SAM ...

Al instalar AWS SAMCLI usando Homebrew, verá el siguiente mensaje de advertencia:

```
Warning: ... AWS will no longer maintain the Homebrew installer for AWS SAM (aws/tap/aws-sam-cli).
```

For AWS supported installations, use the first party installers ...

Causa potencial: ya AWS no se mantiene el Homebrew soporte.

A partir de septiembre de 2023, AWS dejará de mantener el Homebrew instalador del AWS SAMCLI.

Solución: utilice un AWS método de instalación compatible.

- Encontrará los métodos de instalación AWS compatibles en [Instala la AWS SAM CLI](#).

Solución: para seguir utilizándolos Homebrew, utilice el instalador gestionado por la comunidad.

- Puedes utilizar el instalador Homebrew gestionado por la comunidad a su entera discreción. Para obtener instrucciones, consulta [Administración de CLI de AWS SAM con la Homebrew](#).

AWS SAM referencia de conector

Esta sección contiene información de referencia para el tipo de recurso del conector AWS Serverless Application Model (AWS SAM). Para obtener una introducción a los conectores, consulte [Administrar los permisos de recursos con AWS SAM conectores](#).

Tipos de recursos de origen y destino admitidos para los conectores

El tipo de recurso `AWS::Serverless::Connector` admite un número determinado de conexiones entre los recursos de origen y destino. Al configurar los conectores en la AWS SAM plantilla, utilice la siguiente tabla para hacer referencia a las conexiones compatibles y a las propiedades que deben definirse para cada tipo de recurso de origen y destino. Para obtener más información sobre cómo configurar conectores en la plantilla, consulte [AWS::Serverless: :Conector](#).

Para los recursos de origen y destino, cuando se definan en la misma plantilla, utilice la propiedad `Id`. Si lo desea, puedes añadirse un `Qualifier` para reducir el alcance del recurso definido. Cuando el recurso no esté dentro de la misma plantilla, utiliza una combinación de propiedades admitidas.

Para solicitar nuevas conexiones, [envíe una nueva edición](#) al `serverless-application-model` AWS GitHubrepositorio.

Tipo de origen	Tipo de destino	Permisos	Propiedades de origen	Propiedades de destino
AWS::ApiGateway::RestApi	AWS::Lambda::Function	Write	Id o Qualifier , ResourceId y Type	Id o Arn y Type
AWS::ApiGateway::RestApi	AWS::Serverless::Function	Write	Id o Qualifier , ResourceId y Type	Id o Arn y Type
AWS::ApiGatewayV2::Api	AWS::Lambda::Function	Write	Id o Qualifier , ResourceId y Type	Id o Arn y Type
AWS::ApiGatewayV2::Api	AWS::Serverless::Function	Write	Id o Qualifier , ResourceId y Type	Id o Arn y Type
AWS::AppSync::DataSource	AWS::DynamoDB::Table	Read	Id o RoleName y Type	Id o Arn y Type
AWS::AppSync::DataSource	AWS::DynamoDB::Table	Write	Id o RoleName y Type	Id o Arn y Type
AWS::AppSync::DataSource	AWS::Events::EventBus	Write	Id o RoleName y Type	Id o Arn y Type
AWS::AppSync::DataSource	AWS::Lambda::Function	Write	Id o RoleName y Type	Id o Arn y Type

Tipo de origen	Tipo de destino	Permisos	Propiedades de origen	Propiedades de destino
AWS::AppSync::DataSource	AWS::Serverless::Function	Write	Id o RoleName y Type	Id o Arn y Type
AWS::AppSync::DataSource	AWS::Serverless::SimpleTable	Read	Id o RoleName y Type	Id o Arn y Type
AWS::AppSync::DataSource	AWS::Serverless::SimpleTable	Write	Id o RoleName y Type	Id o Arn y Type
AWS::AppSync::GraphQLApi	AWS::Lambda::Function	Write	Id o ResourceId y Type	Id o Arn y Type
AWS::AppSync::GraphQLApi	AWS::Serverless::Function	Write	Id o ResourceId y Type	Id o Arn y Type
AWS::DynamoDB::Table	AWS::Lambda::Function	Read	Id o Arn y Type	Id o RoleName y Type
AWS::DynamoDB::Table	AWS::Serverless::Function	Read	Id o Arn y Type	Id o RoleName y Type
AWS::Events::Rule	AWS::Events::Event Bus	Write	Id o RoleName y Type	Id o Arn y Type
AWS::Events::Rule	AWS::Lambda::Function	Write	Id o Arn y Type	Id o Arn y Type

Tipo de origen	Tipo de destino	Permisos	Propiedades de origen	Propiedades de destino
AWS::Events::Rule	AWS::Serverless::Function	Write	Id o Arn y Type	Id o Arn y Type
AWS::Events::Rule	AWS::Serverless::StateMachine	Write	Id o RoleName y Type	Id o Arn y Type
AWS::Events::Rule	AWS::SNS::Topic	Write	Id o Arn y Type	Id o Arn y Type
AWS::Events::Rule	AWS::SQS::Queue	Write	Id o Arn y Type	Id o Arn, QueueUrl y Type
AWS::Events::Rule	AWS::StepFunctions::StateMachine	Write	Id o RoleName y Type	Id o Arn y Type
AWS::Lambda::Function	AWS::DynamoDB::Table	Read, Write	Id o RoleName y Type	Id o Arn y Type
AWS::Lambda::Function	AWS::Events::EventBus	Write	Id o RoleName y Type	Id o Arn y Type
AWS::Lambda::Function	AWS::Lambda::Function	Write	Id o RoleName y Type	Id o Arn y Type

Tipo de origen	Tipo de destino	Permisos	Propiedades de origen	Propiedades de destino
AWS::Lambda::Function	AWS::Location::PlaceIndex	Read	Id o RoleName y Type	Id o Arn y Type
AWS::Lambda::Function	AWS::S3::Bucket	Read, Write	Id o RoleName y Type	Id o Arn y Type
AWS::Lambda::Function	AWS::Serverless::Function	Write	Id o RoleName y Type	Id o Arn y Type
AWS::Lambda::Function	AWS::Serverless::SimpleTable	Read, Write	Id o RoleName y Type	Id o Arn y Type
AWS::Lambda::Function	AWS::Serverless::StateMachine	Read, Write	Id o RoleName y Type	Id o Arn, Name y Type
AWS::Lambda::Function	AWS::SNS::Topic	Write	Id o RoleName y Type	Id o Arn y Type
AWS::Lambda::Function	AWS::SQS::Queue	Read, Write	Id o RoleName y Type	Id o Arn y Type
AWS::Lambda::Function	AWS::StepFunctions::StateMachine	Read, Write	Id o RoleName y Type	Id o Arn, Name y Type

Tipo de origen	Tipo de destino	Permisos	Propiedades de origen	Propiedades de destino
AWS::S3::Bucket	AWS::Lambda::Function	Write	Id o Arn y Type	Id o Arn y Type
AWS::S3::Bucket	AWS::Serverless::Function	Write	Id o Arn y Type	Id o Arn y Type
AWS::Serverless::Api	AWS::Lambda::Function	Write	Id o Qualifier , ResourceId y Type	Id o Arn y Type
AWS::Serverless::Api	AWS::Serverless::Function	Write	Id o Qualifier , ResourceId y Type	Id o Arn y Type
AWS::Serverless::Function	AWS::DynamoDB::Table	Read, Write	Id o RoleName y Type	Id o Arn y Type
AWS::Serverless::Function	AWS::Events::Event Bus	Write	Id o RoleName y Type	Id o Arn y Type
AWS::Serverless::Function	AWS::Lambda::Function	Write	Id o RoleName y Type	Id o Arn y Type
AWS::Serverless::Function	AWS::S3::Bucket	Read, Write	Id o RoleName y Type	Id o Arn y Type

Tipo de origen	Tipo de destino	Permisos	Propiedades de origen	Propiedades de destino
AWS::Serverless::Function	AWS::Serverless::Function	Write	Id o RoleName y Type	Id o Arn y Type
AWS::Serverless::Function	AWS::Serverless::SimpleTable	Read, Write	Id o RoleName y Type	Id o Arn y Type
AWS::Serverless::Function	AWS::Serverless::StateMachine	Read, Write	Id o RoleName y Type	Id o Arn, Name y Type
AWS::Serverless::Function	AWS::SNS::Topic	Write	Id o RoleName y Type	Id o Arn y Type
AWS::Serverless::Function	AWS::SQS::Queue	Read, Write	Id o RoleName y Type	Id o Arn y Type
AWS::Serverless::Function	AWS::StepFunctions::StateMachine	Read, Write	Id o RoleName y Type	Id o Arn, Name y Type
AWS::Serverless::HttpApi	AWS::Lambda::Function	Write	Id o Qualifier , ResourceId y Type	Id o Arn y Type
AWS::Serverless::HttpApi	AWS::Serverless::Function	Write	Id o Qualifier , ResourceId y Type	Id o Arn y Type

Tipo de origen	Tipo de destino	Permisos	Propiedades de origen	Propiedades de destino
AWS::Serverless::SimpleTable	AWS::Lambda::Function	Read	Id o Arn y Type	Id o RoleName y Type
AWS::Serverless::SimpleTable	AWS::Serverless::Function	Read	Id o Arn y Type	Id o RoleName y Type
AWS::Serverless::StateMachine	AWS::DynamoDB::Table	Read, Write	Id o RoleName y Type	Id o Arn y Type
AWS::Serverless::StateMachine	AWS::Events::Event Bus	Write	Id o RoleName y Type	Id o Arn y Type
AWS::Serverless::StateMachine	AWS::Lambda::Function	Write	Id o RoleName y Type	Id o Arn y Type
AWS::Serverless::StateMachine	AWS::S3::Bucket	Read, Write	Id o RoleName y Type	Id o Arn y Type
AWS::Serverless::StateMachine	AWS::Serverless::Function	Write	Id o RoleName y Type	Id o Arn y Type

Tipo de origen	Tipo de destino	Permisos	Propiedades de origen	Propiedades de destino
AWS::Serverless::StateMachine	AWS::Serverless::SimpleTable	Read, Write	Id o RoleName y Type	Id o Arn y Type
AWS::Serverless::StateMachine	AWS::Serverless::StateMachine	Read, Write	Id o RoleName y Type	Id o Arn, Name y Type
AWS::Serverless::StateMachine	AWS::SNS::Topic	Write	Id o RoleName y Type	Id o Arn y Type
AWS::Serverless::StateMachine	AWS::SQS::Queue	Write	Id o RoleName y Type	Id o Arn y Type
AWS::Serverless::StateMachine	AWS::StepFunctions::StateMachine	Read, Write	Id o RoleName y Type	Id o Arn, Name y Type
AWS::SNS::Topic	AWS::Lambda::Function	Write	Id o Arn y Type	Id o Arn y Type
AWS::SNS::Topic	AWS::Serverless::Function	Write	Id o Arn y Type	Id o Arn y Type

Tipo de origen	Tipo de destino	Permisos	Propiedades de origen	Propiedades de destino
AWS::SNS::Topic	AWS::SQS::Queue	Write	Id o Arn y Type	Id o Arn, QueueUrl y Type
AWS::SQS::Queue	AWS::Lambda::Function	Read, Write	Id o Arn y Type	Id o RoleName y Type
AWS::SQS::Queue	AWS::Serverless::Function	Read, Write	Id o Arn y Type	Id o RoleName y Type
AWS::StepFunctions::StateMachine	AWS::DynamoDB::Table	Read, Write	Id o RoleName y Type	Id o Arn y Type
AWS::StepFunctions::StateMachine	AWS::Events::EventBus	Write	Id o RoleName y Type	Id o Arn y Type
AWS::StepFunctions::StateMachine	AWS::Lambda::Function	Write	Id o RoleName y Type	Id o Arn y Type
AWS::StepFunctions::StateMachine	AWS::S3::Bucket	Read, Write	Id o RoleName y Type	Id o Arn y Type

Tipo de origen	Tipo de destino	Permisos	Propiedades de origen	Propiedades de destino
AWS::Step Functions::StateMachine	AWS::Serverless::Function	Write	Id o RoleName y Type	Id o Arn y Type
AWS::Step Functions::StateMachine	AWS::Serverless::SimpleTable	Read, Write	Id o RoleName y Type	Id o Arn y Type
AWS::Step Functions::StateMachine	AWS::Serverless::StateMachine	Read, Write	Id o RoleName y Type	Id o Arn, Name y Type
AWS::Step Functions::StateMachine	AWS::SNS::Topic	Write	Id o RoleName y Type	Id o Arn y Type
AWS::Step Functions::StateMachine	AWS::SQS::Queue	Write	Id o RoleName y Type	Id o Arn y Type
AWS::Step Functions::StateMachine	AWS::Step Functions::StateMachine	Read, Write	Id o RoleName y Type	Id o Arn, Name y Type

Políticas de IAM creadas por conectores

En esta sección se documentan las políticas AWS Identity and Access Management (de IAM) que se crean AWS SAM al utilizar los conectores.

De AWS::DynamoDB::Table a AWS::Lambda::Function

Tipo de política

[La política administrada por el cliente](#) asociada al rol de AWS::Lambda::Function.

Categorías de acceso

Read

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "dynamodb:DescribeStream",
        "dynamodb:GetRecords",
        "dynamodb:GetShardIterator",
        "dynamodb:ListStreams"
      ],
      "Resource": [
        "%{Source.Arn}/stream/*"
      ]
    }
  ]
}
```

De AWS::Events::Rule a AWS::SNS::Topic

Tipo de política

[AWS::SNS::TopicPolicy](#) asociada a la AWS::SNS::Topic.

Categorías de acceso

Write

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "events.amazonaws.com"
      },
```



```
    "Resource": "%{Destination.Arn}",
    "Action": "sns:Publish",
    "Condition": {
      "ArnEquals": {
        "aws:SourceArn": "%{Source.Arn}"
      }
    }
  }
]
```

De `AWS::Events::Rule` a `AWS::Events::EventBus`

Tipo de política

[La política administrada por el cliente](#) asociada al rol de `AWS::Events::Rule`.

Categorías de acceso

Write

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "events:PutEvents"
      ],
      "Resource": [
        "%{Destination.Arn}"
      ]
    }
  ]
}
```

De `AWS::Events::Rule` a `AWS::StepFunctions::StateMachine`

Tipo de política

[La política administrada por el cliente](#) asociada al rol de `AWS::Events::Rule`.

Categorías de acceso

Write

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "states:StartExecution"
      ],
      "Resource": [
        "%{Destination.Arn}"
      ]
    }
  ]
}
```

De `AWS::Events::Rule` a `AWS::Lambda::Function`

Tipo de política

[AWS::Lambda::Permission](#) asociada a la `AWS::Lambda::Function`.

Categorías de acceso

Write

```
{
  "Action": "lambda:InvokeFunction",
  "Principal": "events.amazonaws.com",
  "SourceArn": "%{Source.Arn}"
}
```

De `AWS::Events::Rule` a `AWS::SQS::Queue`

Tipo de política

[AWS::SQS::QueuePolicy](#) asociada a la `AWS::SQS::Queue`.

Categorías de acceso

Write

```
{
  "Statement": [
    {
```

```
"Effect": "Allow",
"Principal": {
  "Service": "events.amazonaws.com"
},
"Resource": "%{Destination.Arn}",
"Action": "sqs:SendMessage",
"Condition": {
  "ArnEquals": {
    "aws:SourceArn": "%{Source.Arn}"
  }
}
}
```

De `AWS::Lambda::Function` a `AWS::Lambda::Function`

Tipo de política

[La política administrada por el cliente](#) asociada al rol de `AWS::Lambda::Function`.

Categorías de acceso

Write

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "lambda:InvokeAsync",
        "lambda:InvokeFunction"
      ],
      "Resource": [
        "%{Destination.Arn}"
      ]
    }
  ]
}
```

De `AWS::Lambda::Function` a `AWS::S3::Bucket`

Tipo de política

[La política administrada por el cliente](#) asociada al rol de `AWS::Lambda::Function`.

Categorías de acceso

Read

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:GetObjectAcl",
        "s3:GetObjectLegalHold",
        "s3:GetObjectRetention",
        "s3:GetObjectTorrent",
        "s3:GetObjectVersion",
        "s3:GetObjectVersionAcl",
        "s3:GetObjectVersionForReplication",
        "s3:GetObjectVersionTorrent",
        "s3:ListBucket",
        "s3:ListBucketMultipartUploads",
        "s3:ListBucketVersions",
        "s3:ListMultipartUploadParts"
      ],
      "Resource": [
        "%{Destination.Arn}",
        "%{Destination.Arn}/*"
      ]
    }
  ]
}
```

Write

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:AbortMultipartUpload",
        "s3:DeleteObject",
        "s3:DeleteObjectVersion",

```

```

        "s3:PutObject",
        "s3:PutObjectLegalHold",
        "s3:PutObjectRetention",
        "s3:RestoreObject"
    ],
    "Resource": [
        "%{Destination.Arn}",
        "%{Destination.Arn}/*"
    ]
}
]
}

```

De AWS::Lambda::Function a AWS::DynamoDB::Table

Tipo de política

[La política administrada por el cliente](#) asociada al rol de AWS::Lambda::Function.

Categorías de acceso

Read

```

{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "dynamodb:GetItem",
        "dynamodb:Query",
        "dynamodb:Scan",
        "dynamodb:BatchGetItem",
        "dynamodb:ConditionCheckItem",
        "dynamodb: PartiQLSelect"
      ],
      "Resource": [
        "%{Destination.Arn}",
        "%{Destination.Arn}/index/*"
      ]
    }
  ]
}

```

Write

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "dynamodb:PutItem",
        "dynamodb:UpdateItem",
        "dynamodb>DeleteItem",
        "dynamodb:BatchWriteItem",
        "dynamodb: PartiQLDelete",
        "dynamodb: PartiQLInsert",
        "dynamodb: PartiQLUpdate"
      ],
      "Resource": [
        "%{Destination.Arn}",
        "%{Destination.Arn}/index/*"
      ]
    }
  ]
}
```

De AWS::Lambda::Function a AWS::SQS::Queue

Tipo de política

[La política administrada por el cliente](#) asociada al rol de AWS::Lambda::Function.

Categorías de acceso

Read

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "sqs:ReceiveMessage",
        "sqs:GetQueueAttributes"
      ],
      "Resource": [
        "%{Destination.Arn}"
      ]
    }
  ]
}
```

```
]
}
```

Write

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "sqs:DeleteMessage",
        "sqs:SendMessage",
        "sqs:ChangeMessageVisibility",
        "sqs:PurgeQueue"
      ],
      "Resource": [
        "%{Destination.Arn}"
      ]
    }
  ]
}
```

De `AWS::Lambda::Function` a `AWS::SNS::Topic`

Tipo de política

[La política administrada por el cliente](#) asociada al rol de `AWS::Lambda::Function`.

Categorías de acceso

Write

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "sns:Publish"
      ],
      "Resource": [
        "%{Destination.Arn}"
      ]
    }
  ]
}
```

```
]
}
```

De AWS::Lambda::Function a AWS::StepFunctions::StateMachine

Tipo de política

[La política administrada por el cliente](#) asociada al rol de AWS::Lambda::Function.

Categorías de acceso

Write

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "states:StartExecution",
        "states:StartSyncExecution"
      ],
      "Resource": [
        "%{Destination.Arn}"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "states:StopExecution"
      ],
      "Resource": [
        "arn:${AWS::Partition}:states:${AWS::Region}:${AWS::AccountId}:execution:
        %{Destination.Name}:*"
      ]
    }
  ]
}
```

Read

```
{
  "Statement": [
    {
      "Effect": "Allow",
```



```

    "Action": [
      "states:DescribeStateMachine",
      "states:ListExecutions"
    ],
    "Resource": [
      "%{Destination.Arn}"
    ]
  },
  {
    "Effect": "Allow",
    "Action": [
      "states:DescribeExecution",
      "states:DescribeStateMachineForExecution",
      "states:GetExecutionHistory"
    ],
    "Resource": [
      "arn:${AWS::Partition}:states:${AWS::Region}:${AWS::AccountId}:execution:
%{Destination.Name}:*"
    ]
  }
]
}

```

De `AWS::Lambda::Function` a `AWS::Events::EventBus`

Tipo de política

[La política administrada por el cliente](#) asociada al rol de `AWS::Lambda::Function`.

Categorías de acceso

Write

```

{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "events:PutEvents"
      ],
      "Resource": [
        "%{Destination.Arn}"
      ]
    }
  ]
}

```

```
]
}
```

De `AWS::Lambda::Function` a `AWS::Location::PlaceIndex`

Tipo de política

[La política administrada por el cliente](#) asociada al rol de `AWS::Lambda::Function`.

Categorías de acceso

Read

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "geo:DescribePlaceIndex",
        "geo:GetPlace",
        "geo:SearchPlaceIndexForPosition",
        "geo:SearchPlaceIndexForSuggestions",
        "geo:SearchPlaceIndexForText"
      ],
      "Resource": [
        "%{Destination.Arn}"
      ]
    }
  ]
}
```

De `AWS::ApiGatewayV2::Api` a `AWS::Lambda::Function`

Tipo de política

[AWS::Lambda::Permission](#) asociada a la `AWS::Lambda::Function`.

Categorías de acceso

Write

```
{
  "Action": "lambda:InvokeFunction",
  "Principal": "apigateway.amazonaws.com",
```

```
"SourceArn": "arn:${AWS::Partition}:execute-api:${AWS::Region}:${AWS::AccountId}:
${Source.ResourceId}/${Source.Qualifier}"
}
```

De `AWS::ApiGateway::RestApi` a `AWS::Lambda::Function`

Tipo de política

[AWS::Lambda::Permission](#) asociada a la `AWS::Lambda::Function`.

Categorías de acceso

Write

```
{
  "Action": "lambda:InvokeFunction",
  "Principal": "apigateway.amazonaws.com",
  "SourceArn": "arn:${AWS::Partition}:execute-api:${AWS::Region}:${AWS::AccountId}:
${Source.ResourceId}/${Source.Qualifier}"
}
```

De `AWS::SNS::Topic` a `AWS::SQS::Queue`

Tipo de política

[AWS::SQS::QueuePolicy](#) asociada a la `AWS::SQS::Queue`.

Categorías de acceso

Write

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "sns.amazonaws.com"
      },
      "Resource": "%{Destination.Arn}",
      "Action": "sqs:SendMessage",
      "Condition": {
        "ArnEquals": {
          "aws:SourceArn": "%{Source.Arn}"
        }
      }
    }
  ]
}
```

```
}  
}  
]  
}
```

De `AWS::SNS::Topic` a `AWS::Lambda::Function`

Tipo de política

[AWS::Lambda::Permission](#) asociada a la `AWS::Lambda::Function`.

Categorías de acceso

Write

```
{  
  "Action": "lambda:InvokeFunction",  
  "Principal": "sns.amazonaws.com",  
  "SourceArn": "%{Source.Arn}"  
}
```

De `AWS::SQS::Queue` a `AWS::Lambda::Function`

Tipo de política

[La política administrada por el cliente](#) asociada al rol de `AWS::Lambda::Function`.

Categorías de acceso

Write

```
{  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": [  
        "sqs:DeleteMessage"  
      ],  
      "Resource": [  
        "%{Source.Arn}"  
      ]  
    }  
  ]  
}
```

Read

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "sqs:ReceiveMessage",
        "sqs:GetQueueAttributes"
      ],
      "Resource": [
        "%{Source.Arn}"
      ]
    }
  ]
}
```

De `AWS::S3::Bucket` a `AWS::Lambda::Function`

Tipo de política

[AWS::Lambda::Permission](#) asociada a la `AWS::Lambda::Function`.

Categorías de acceso

Write

```
{
  "Action": "lambda:InvokeFunction",
  "Principal": "s3.amazonaws.com",
  "SourceArn": "%{Source.Arn}",
  "SourceAccount": "${AWS::AccountId}"
}
```

De `AWS::StepFunctions::StateMachine` a `AWS::Lambda::Function`

Tipo de política

[La política administrada por el cliente](#) asociada al rol de `AWS::StepFunctions::StateMachine`.

Categorías de acceso

Write

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "lambda:InvokeAsync",
        "lambda:InvokeFunction"
      ],
      "Resource": [
        "%{Destination.Arn}"
      ]
    }
  ]
}
```

De `AWS::StepFunctions::StateMachine` a `AWS::SNS::Topic`

Tipo de política

[La política administrada por el cliente](#) asociada al rol de `AWS::StepFunctions::StateMachine`.

Categorías de acceso

Write

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "sns:Publish"
      ],
      "Resource": [
        "%{Destination.Arn}"
      ]
    }
  ]
}
```

De `AWS::StepFunctions::StateMachine` a `AWS::SQS::Queue`

Tipo de política

[La política administrada por el cliente](#) asociada al rol de `AWS::StepFunctions::StateMachine`.

Categorías de acceso

Write

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "sqs:SendMessage"
      ],
      "Resource": [
        "%{Destination.Arn}"
      ]
    }
  ]
}
```

De `AWS::StepFunctions::StateMachine` a `AWS::S3::Bucket`

Tipo de política

[La política administrada por el cliente](#) asociada al rol de `AWS::StepFunctions::StateMachine`.

Categorías de acceso

Read

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:GetObjectAcl",
        "s3:GetObjectLegalHold",
        "s3:GetObjectRetention",
        "s3:GetObjectTorrent",

```

```

        "s3:GetObjectVersion",
        "s3:GetObjectVersionAcl",
        "s3:GetObjectVersionForReplication",
        "s3:GetObjectVersionTorrent",
        "s3:ListBucket",
        "s3:ListBucketMultipartUploads",
        "s3:ListBucketVersions",
        "s3:ListMultipartUploadParts"
    ],
    "Resource": [
        "%{Destination.Arn}",
        "%{Destination.Arn}/*"
    ]
}
]
}

```

Write

```

{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:AbortMultipartUpload",
        "s3:DeleteObject",
        "s3:DeleteObjectVersion",
        "s3:PutObject",
        "s3:PutObjectLegalHold",
        "s3:PutObjectRetention",
        "s3:RestoreObject"
      ],
      "Resource": [
        "%{Destination.Arn}",
        "%{Destination.Arn}/*"
      ]
    }
  ]
}

```

De `AWS::StepFunctions::StateMachine` a `AWS::DynamoDB::Table`

Tipo de política

[La política administrada por el cliente](#) asociada al rol de `AWS::StepFunctions::StateMachine`.

Categorías de acceso

Read

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "dynamodb:GetItem",
        "dynamodb:Query",
        "dynamodb:Scan",
        "dynamodb:BatchGetItem",
        "dynamodb:ConditionCheckItem",
        "dynamodb: PartiQLSelect"
      ],
      "Resource": [
        "%{Destination.Arn}",
        "%{Destination.Arn}/index/*"
      ]
    }
  ]
}
```

Write

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "dynamodb:PutItem",
        "dynamodb:UpdateItem",
        "dynamodb>DeleteItem",
        "dynamodb:BatchWriteItem",
        "dynamodb: PartiQLDelete",
        "dynamodb: PartiQLInsert",
        "dynamodb: PartiQLUpdate"
      ],
      "Resource": [
```

```

        "%{Destination.Arn}",
        "%{Destination.Arn}/index/*"
    ]
}
]
}

```

De `AWS::StepFunctions::StateMachine` a `AWS::StepFunctions::StateMachine`

Tipo de política

[La política administrada por el cliente](#) asociada al rol de `AWS::StepFunctions::StateMachine`.

Categorías de acceso

Read

```

{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "states:DescribeExecution"
      ],
      "Resource": [
        "arn:${AWS::Partition}:states:${AWS::Region}:${AWS::AccountId}:execution:
%{Destination.Name}:*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "events:DescribeRule"
      ],
      "Resource": [
        "arn:${AWS::Partition}:events:${AWS::Region}:${AWS::AccountId}:rule/
StepFunctionsGetEventsForStepFunctionsExecutionRule"
      ]
    }
  ]
}

```

Write

```

{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "states:StartExecution"
      ],
      "Resource": [
        "%{Destination.Arn}"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "states:StopExecution"
      ],
      "Resource": [
        "arn:${AWS::Partition}:states:${AWS::Region}:${AWS::AccountId}:execution:
%{Destination.Name}:*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "events:PutTargets",
        "events:PutRule"
      ],
      "Resource": [
        "arn:${AWS::Partition}:events:${AWS::Region}:${AWS::AccountId}:rule/
StepFunctionsGetEventsForStepFunctionsExecutionRule"
      ]
    }
  ]
}

```

De `AWS::StepFunctions::StateMachine` a `AWS::Events::EventBus`

Tipo de política

[La política administrada por el cliente](#) asociada al rol de `AWS::StepFunctions::StateMachine`.

Categorías de acceso

Write

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "events:PutEvents"
      ],
      "Resource": [
        "%{Destination.Arn}"
      ]
    }
  ]
}
```

De `AWS::AppSync::DataSource` a `AWS::DynamoDB::Table`

Tipo de política

[La política administrada por el cliente](#) asociada al rol de `AWS::AppSync::DataSource`.

Categorías de acceso

Read

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "dynamodb:GetItem",
        "dynamodb:Query",
        "dynamodb:Scan",
        "dynamodb:BatchGetItem",
        "dynamodb:ConditionCheckItem",
        "dynamodb: PartiQLSelect"
      ],
      "Resource": [
        "%{Destination.Arn}",
        "%{Destination.Arn}/index/*"
      ]
    }
  ]
}
```

```
]
}
```

Write

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "dynamodb:PutItem",
        "dynamodb:UpdateItem",
        "dynamodb>DeleteItem",
        "dynamodb:BatchWriteItem",
        "dynamodb: PartiQLDelete",
        "dynamodb: PartiQLInsert",
        "dynamodb: PartiQLUpdate"
      ],
      "Resource": [
        "%{Destination.Arn}",
        "%{Destination.Arn}/index/*"
      ]
    }
  ]
}
```

De `AWS::AppSync::DataSource` a `AWS::Lambda::Function`

Tipo de política

[La política administrada por el cliente](#) asociada al rol de `AWS::AppSync::DataSource`.

Categorías de acceso

Write

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "lambda:InvokeAsync",
        "lambda:InvokeFunction"
      ]
    }
  ]
}
```

```
    ],
    "Resource": [
        "%{Destination.Arn}",
        "%{Destination.Arn}:"
    ]
}
]
```

De `AWS::AppSync::DataSource` a `AWS::Events::EventBus`

Tipo de política

[La política administrada por el cliente](#) asociada al rol de `AWS::AppSync::DataSource`.

Categorías de acceso

Write

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "events:PutEvents"
      ],
      "Resource": [
        "%{Destination.Arn}"
      ]
    }
  ]
}
```

De `AWS::AppSync::GraphQLApi` a `AWS::Lambda::Function`

Tipo de política

[AWS::Lambda::Permission](#) asociada a la `AWS::Lambda::Function`.

Categorías de acceso

Write

```
{
  "Action": "lambda:InvokeFunction",
  "Principal": "appsync.amazonaws.com",
  "SourceArn": "arn:${AWS::Partition}:appsync:${AWS::Region}:${AWS::AccountId}:apis/
%{Source.ResourceId}"
}
```

Instalación de Docker para usarlo con la de AWS SAM CLI

Docker es una aplicación que ejecuta contenedores en su máquina. Con Docker, AWS SAM puede proporcionar un entorno local similar a AWS Lambda un contenedor para crear, probar y depurar sus aplicaciones sin servidor.

Note

Docker solo es necesario para probar las aplicaciones de forma local y para crear paquetes de implementación mediante la opción `--use-container`.

Temas

- [Instalación Docker](#)
- [Siguiendo pasos](#)

Instalación Docker

Siga estas instrucciones para instalar Docker en su sistema operativo.

Linux

Docker está disponible en muchos sistemas operativos diferentes, incluidas las distribuciones de Linux más modernas, como CentOS, Debian o Ubuntu. Para obtener más información sobre cómo instalar Docker en su sistema operativo concreto, consulte [Conseguir Docker](#) en el sitio web de Docker Docs.

Para instalar Docker en Amazon Linux 2 o Amazon Linux 2023

1. Actualice la caché de paquetes y los paquetes instalados en la instancia.

```
$ sudo yum update -y
```

2. Instale el paquete de Community Edition de Docker más reciente.

- Para Amazon Linux 2, ejecute lo siguiente:

```
$ sudo amazon-linux-extras install docker
```

- Para Amazon Linux 2023, ejecute lo siguiente:

```
$ sudo yum install -y docker
```

3. Inicie el servicio Docker.

```
$ sudo service docker start
```

4. Agregue el `ec2-user` al grupo `docker` para que pueda ejecutar comandos de Docker sin usar `sudo`.

```
$ sudo usermod -a -G docker ec2-user
```

5. Cierre sesión y vuelva a iniciarla para actualizar los nuevos permisos de grupo de `docker`. Para ello, cierre la ventana de su terminal de SSH actual y vuelva a conectarse a su instancia en una ventana nueva. De esta forma, la nueva sesión de SSH debería tener los permisos de grupo de `docker` adecuados.

6. Compruebe que el `ec2-user` puede ejecutar comandos de Docker sin `sudo`.

```
$ docker ps
```

Debería ver el siguiente resultado, lo que confirma que Docker está instalado y en ejecución:

CONTAINER ID	IMAGE	COMMAND	CREATED
STATUS	PORTS	NAMES	

Note

En Linux, para compilar y ejecutar funciones de eventos Lambda con una arquitectura de conjunto de instrucciones diferente a la de su máquina host, hay pasos adicionales de

configuración. Docker Por ejemplo, para ejecutar funciones arm64 en una unidad x86_64, puede ejecutar el siguiente comando para configurar el daemon de Docker: `docker run --rm --privileged multiarch/qemu-user-static --reset -p yes`.

Si tiene problemas para instalar Docker, consulte [Solución de errores de la instalación](#). O bien, consulte la sección de [solución](#) de problemas con Post-installation los pasos para Linux en el sitio web de Docker Docs.

macOS

Note

Docker Desktop cuenta con soporte oficial, pero a partir de la versión 1.47.0 de la CLI de AWS SAM, puede usar alternativas siempre que usen el tiempo de ejecución de Docker.

1. Instalar Docker

La CLI de AWS SAM es compatible con Docker para macOS Sierra 10.12 o posterior. Para saber cómo instalar Docker, consulte [Instalar Docker Desktop para Mac](#) en el sitio web de Docker Docs.

2. Configuración de las unidades compartidas

AWS SAMCLI Requiere que el directorio del proyecto, o cualquier directorio principal, aparezca en una unidad compartida. Para ello puede ser necesario el uso compartido de archivos. Para obtener más información, consulte el tema de solución de problemas [El montaje de volúmenes requiere el uso compartido de archivos](#) en Docker Docs.

3. Verificar la instalación

Una vez Docker esté instalado, compruebe que funciona. Confirme también que puede ejecutar comandos de Docker desde la línea de comandos (por ejemplo, `docker ps`). No necesita instalar, recuperar ni extraer ningún contenedor; la CLI de AWS SAM lo hace automáticamente según sea necesario.

Si tiene problemas con la instalación de Docker, consulte la sección [Solución de problemas y diagnóstico](#) del sitio web de Docker Docs para más consejos sobre resolución de problemas.

Windows

Note

AWS SAM oficialmente es compatible con Docker Desktop. Sin embargo, a partir de la versión 1.47.0 de la CLI de AWS SAM, puede usar alternativas siempre que usen el tiempo de ejecución de Docker.

1. Instale Docker.

Docker Desktop es compatible con el sistema operativo Windows más reciente. Para las versiones antiguas de Windows, está disponible Docker Toolbox. Elija su versión de Windows para instalar Docker paso a paso de manera correcta:

- Para instalar Docker para Windows 10, consulte [Instalar Docker Desktop para Windows](#) en el sitio web de Docker Docs.
- Para realizar la instalación Docker para versiones anteriores de Windows, consulte [The Docker Toolbox](#) en el repositorio de Docker Toolbox. GitHub

2. Configuración de las unidades compartidas

AWS SAM CLI requiere que el directorio del proyecto, o cualquier directorio principal, esté incluido en una unidad compartida. En algunos casos, debe compartir la unidad para que Docker funcione correctamente.

3. Verifica la instalación.

Una vez Docker esté instalado, compruebe que funciona. Confirme también que puede ejecutar comandos de Docker desde la línea de comandos (por ejemplo, `docker ps`). No necesita instalar, recuperar ni extraer ningún contenedor; la CLI de AWS SAM lo hace automáticamente según sea necesario.

Si tiene problemas con la instalación de Docker, consulte la sección [Solución de problemas y diagnóstico](#) del sitio web de Docker Docs para más consejos sobre resolución de problemas.

Siguientes pasos

Para saber cómo instalar el AWS SAM CLI, consulte [Instala la AWS SAM CLI](#).

Instalación Finch para usar con el AWS SAM CLI

Finch AWS es una herramienta de desarrollo de contenedores que ofrece una alternativa Docker, especialmente beneficiosa en los sistemas macOS y Linux. Para obtener más información, consulte [Finch en GitHub](#). Con Finch, AWS SAM puede proporcionar un entorno local similar a AWS Lambda un contenedor para crear, probar y depurar sus aplicaciones sin servidor.

Note

Necesita Finch para probar las aplicaciones de forma local, para crear paquetes de implementación mediante la opción `--use-container` y para crear funciones de imagen de OCI. A partir de la versión 1.145.0 de la AWS SAM CLI, la AWS SAM CLI es compatible con Finch el tiempo de ejecución de contenedores alternativo a. Docker

Temas

- [Instalación Finch](#)
- [Configuración Finch por AWS SAM CLI](#)
- [Verificar la instalación](#)
- [Detención Finch](#)
- [Resolución de problemas](#)
- [Siguiendo pasos](#)

Instalación Finch

Siga estas instrucciones para instalar Finch en su sistema operativo.

Requisitos previos

Note

Finch la integración en AWS SAM CLI está disponible en los sistemas macOS y Linux. Para los usuarios de Windows, evalúe la posibilidad de utilizar una máquina virtual o un entorno Linux para la funcionalidad de Finch.

Versión para macOS:

- macOS 15 Sequoia
- macOS 14 Sonoma

Note

Finch es compatible con las dos últimas versiones de macOS.

Linux:

- Linux Kernel v4.x+

Instalación de Finch

macOS

Para instalar Finch en macOS

1. Instale Finch mediante la CLI de brew. Si se le solicita, escriba su contraseña de macOS.

```
$ brew install finch
```

2. Compruebe la versión Finch instalada.

```
$ finch --version
```

3. Inicialice la máquina virtual de Finch (configuración inicial).

```
$ finch vm init
```

4. Inicie la máquina virtual de Finch.

```
$ finch vm start
```

5. Compruebe el estado de la máquina virtual de Finch.

```
$ finch vm status
```

6. Compruebe que Finch se ejecute correctamente.

```
$ finch --info
$ finch ps
```

Para obtener más información, consulte la [Instalación de Finch en macOS](#).

Linux

Puede instalar Finch en Amazon Linux, Ubuntu o mediante un método de instalación genérico. Para obtener más información, consulte la [Instalación de Finch en Linux](#).

Configuración Finch por AWS SAM CLI

Tras la instalación de Finch, la AWS SAM CLI detecta y utiliza automáticamente Finch como tiempo de ejecución del contenedor cuando Docker no se ejecuta.

Note

Finch sirve como alternativa para Docker. Si tiene ambos Finch instalados Docker y en ejecución, la AWS SAM CLI prioriza de forma Docker predeterminada.

Configuración de preferencias de administrador (macOS)

Puede configurar la AWS SAM CLI para que utilice un tiempo de ejecución de contenedor específico como predeterminado mediante la creación de un archivo de preferencias del administrador. Esta preferencia de todo el sistema anula el comportamiento alternativo predeterminado.

Métodos de configuración disponibles

- Definir Finch como tiempo de ejecución de contenedor preferido:

```
$ sudo /usr/libexec/PlistBuddy -c "Add :DefaultContainerRuntime string finch" /
Library/Preferences/com.amazon.samcli.plist
```

- Definir Docker como tiempo de ejecución de contenedor preferido:

```
$ sudo /usr/libexec/PlistBuddy -c "Add :DefaultContainerRuntime string docker" /
Library/Preferences/com.amazon.samcli.plist
```

- Eliminar la preferencia (volver al comportamiento predeterminado):

```
$ sudo /usr/libexec/PlistBuddy -c "Remove :DefaultContainerRuntime" /Library/Preferences/com.amazon.samcli.plist
```

Note

Cuando no se establece ninguna preferencia de administrador, la AWS SAM CLI la utiliza de forma Docker predeterminada cuando está disponible, y Finch como alternativa cuando no Docker se está ejecutando.

Diferencias de montaje de directorios (macOS)

En macOS, los comandos AWS SAM locales fallan si ubicas el proyecto fuera del directorio principal (~) o/Volumes. Este diseño restringe el montaje de directorios automático por motivos de seguridad sin su consentimiento.

Opciones de mitigación

- Opción 1: abra la configuración de Finch en un editor de texto ~/.finch/finch.yaml y agregue la ruta del proyecto en la sección `additional_directories`, y luego reinicie la máquina virtual de Finch. Para obtener información adicional, consulte la [documentación de Finch](#).

```
$ finch vm stop  
$ finch vm start
```

- Opción 2: mueva el proyecto a su directorio principal.

Verificar la instalación

Después de instalarlo e iniciarlo Finch, compruebe que funciona con AWS SAM CLI.

Verificación básica

Ejecute el siguiente comando para comprobar que la AWS SAM CLI puede utilizar Finch:

```
$ sam build --use-container
```

Debería ver cómo se utiliza la AWS SAM CLI Finch para crear la aplicación en un entorno de contenedores.

 Note

Finch sirve como alternativa para Docker. Si Docker se está ejecutando, la AWS SAM CLI priorizará Docker de forma predeterminada.

Testeo Finch como alternativa

Para probar Finch como alternativa cuando Docker se está ejecutando, primero debe detener Docker.

Para probar Finch funcionalidad alternativa

1. Detenga Docker con los comandos apropiados para el sistema operativo.

macOS:

```
$ docker stop $(docker ps -q)
$ pkill -f "Docker Desktop"
```

Verifique que Docker esté detenido.

```
$ docker ps
```

Linux:

```
$ docker stop $(docker ps -q)
$ sudo systemctl stop docker
```

Verifique que Docker esté detenido.

```
$ sudo systemctl status docker
```

2. Ejecute el `sam build --use-container` comando para verificar que la AWS SAM CLI Finch lo utilice automáticamente como respaldo.

Detención Finch

Si necesita detener Finch, utilice los siguientes comandos:

macOS:

```
$ finch vm stop  
$ sudo pkill -f finch
```

Verifique que Finch esté detenido.

```
$ finch vm status
```

Linux:

```
$ sudo systemctl stop finch  
$ sudo pkill -f finch
```

Verifique que Finch esté detenido.

```
$ sudo systemctl status finch
```

Resolución de problemas

Si tiene problemas con Finch:

- Compruebe que el sistema operativo cumpla los requisitos previos.
- Compruebe que ha habilitado la virtualización en el sistema.
- Para macOS, compruebe que tiene suficiente espacio en disco para la máquina virtual.
- Para Linux, compruebe que tiene los permisos adecuados para ejecutar los comandos `systemctl`.
- Si encuentra errores de CloudFormation implementación con errores de «tipo de medio no compatible» al insertar imágenes de varias arquitecturas en Amazon Elastic Container Registry (Amazon ECR), cree un único artefacto de imagen en lugar de utilizar imágenes de varias arquitecturas. Esto ocurre porque Finch empuja el índice de la imagen y la imagen a Amazon ECR, mientras que Docker solo empuja la imagen.
- Si la creación entre plataformas falla porque la arquitectura de la función de destino no coincide con la arquitectura de la máquina host en Linux, habilite la emulación entre plataformas mediante


```
la ejecución sudo finch run --privileged --rm tonistiigi/binfmt:master --  
install all.
```

Para obtener información adicional sobre la solución de problemas, consulte la [documentación de Finch](#) o la [AWS SAM CLI solución de problemas](#).

Siguientes pasos

Ahora que lo ha Finch instalado, puede usarlo con AWS SAM CLI para:

- Crear aplicaciones sin servidor mediante contenedores con `sam build --use-container`. Para obtener más información, consulte [the section called “compilación de sam”](#).
- Pruebe las funciones de eventos de Lambda localmente con `sam local invoke`. Para obtener más información, consulte [the section called “invocar sam local”](#).
- Iniciar una API Gateway local con `sam local start-api`. Para obtener más información, consulte [the section called “sam local start-api”](#).
- Iniciar un punto de conexión de Lambda local con `sam local start-lambda`. Para obtener más información, consulte [the section called “sam local start-lambda”](#).

Para obtener más información sobre el uso de AWS SAM CLI con contenedores, consulte [Compilación predeterminada con AWS SAM](#).

Repositorios de imágenes para AWS SAM

AWS SAM simplifica las tareas de integración continua y entrega continua (CI/CD) para aplicaciones sin servidor con la ayuda de crear imágenes de contenedores. Las imágenes que AWS SAM proporciona incluyen la interfaz de línea de AWS SAM comandos (CLI) y las herramientas de compilación para varios tiempos de ejecución compatibles AWS Lambda . Esto facilita la creación y el empaquetado de aplicaciones sin servidor mediante la CLI de AWS SAM. Puede usar estas imágenes con CI/CD sistemas para automatizar la creación y el despliegue de AWS SAM aplicaciones. Para ver ejemplos, consulte [Implemente con CI/CD sistemas y canalizaciones](#).

AWS SAM Los URI de las imágenes del contenedor de compilación se etiquetan con la versión de la AWS SAMCLI incluida en esa imagen. Si especificas el URI sin etiqueta, se utiliza la versión más reciente. Por ejemplo, `public.ecr.aws/sam/build-nodejs20.x` utiliza la imagen más reciente. Sin embargo, `public.ecr.aws/sam/build-nodejs20.x:1.24.1` utiliza la imagen que contiene la versión 1.24.1 de la CLI de AWS SAM .

A partir de la versión 1.33.0 de AWS SAMCLI, tanto x86_64 las imágenes como las de arm64 contenedor están disponibles para los tiempos de ejecución compatibles. Para obtener más información, consulta [Tiempos de ejecución de Lambda](#) en la Guía para desarrolladores de AWS Lambda .

Note

Antes de la versión 1.22.0 de AWS SAMCLI, DockerHub era el repositorio predeterminado del que se AWS SAMCLI extraía la imagen del contenedor. A partir de la versión 1.22.0, el repositorio por defecto se cambió a Amazon Elastic Container Registry Public (Amazon ECR Public). Para extraer una imagen de contenedor de un repositorio que no sea el predeterminado actual, puede usar el comando [compilación de sam](#) con la opción `--build-image`. Los ejemplos que aparecen al final de este tema muestran cómo crear aplicaciones con imágenes de DockerHub repositorio.

URI del repositorio de imágenes

En la siguiente tabla se enumeran los URI de las imágenes de contenedores de [compilación públicos de](#) Amazon ECR que puede usar para crear y empaquetar aplicaciones sin servidor. AWS SAM

Note

Amazon ECR Public reemplazó a DockerHub partir de la versión 1.22.0 de la CLI de AWS SAM. Si utiliza una versión anterior de AWS SAMCLI, le recomendamos que la actualice.

Tiempo de ejecución	Amazon ECR Public
Tiempo de ejecución personalizado (AL2023)	public.ecr.aws/sam/build-provided.al2023
Tiempo de ejecución personalizado (AL2)	public.ecr.aws/sam/build-provided.al2
Tiempo de ejecución personalizado	public.ecr.aws/sam/proporcionado por la compilación
Java 25	public.ecr.aws/sam/build-java25

Tiempo de ejecución	Amazon ECR Public
Java 21	public.ecr.aws/sam/build-java21
Java 17	public.ecr.aws/sam/build-java17
Java 17 (AL2023)	public.ecr.aws/sam/build-java17.al.2023
Java 11	public.ecr.aws/sam/build-java11
Java 11 (AL2023)	public.ecr.aws/sam/build-java11.al.2023
Java 8 (AL2023)	public.ecr.aws/sam/build-java8.al.2023
Java 8	public.ecr.aws/sam/build-java8
.NET 9	public.ecr.aws/sam/build-dotnet9
.NET 8	public.ecr.aws/sam/build-dotnet8
.NET 7	public.ecr.aws/sam/build-dotnet7
.NET 6	public.ecr.aws/sam/build-dotnet6
Node.js 24	public.ecr.aws/sam/build-nodejs24.x
Node.js 22	public.ecr.aws/sam/build-nodejs22.x
Node.js 20	public.ecr.aws/sam/build-nodejs20.x
Node.js 18	public.ecr.aws/sam/build-nodejs18.x
Node.js 16	public.ecr.aws/sam/build-nodejs16.x
Python 3.14	public.ecr.aws/sam/build-python3.14
Python 3.13	public.ecr.aws/sam/build-python3.13
Python 3.12	public.ecr.aws/sam/build-python3.12
Python 3.11	public.ecr.aws/sam/build-python3.11

Tiempo de ejecución	Amazon ECR Public
Python 3.10	public.ecr.aws/sam/build-python3.10
Python 3.9	public.ecr.aws/sam/build-python3.9
Python 3.8	public.ecr.aws/sam/build-python3.8
Ruby 4.0	public.ecr.aws/sam/build-ruby4.0
Ruby 3.4	public.ecr.aws/sam/build-ruby3.4
Ruby 3.3	public.ecr.aws/sam/build-ruby3.3
Ruby 3.2	public.ecr.aws/sam/build-ruby3.2

Ejemplos

Los siguientes dos comandos de muestra crean aplicaciones mediante imágenes de contenedores del repositorio de imágenes:

Crea una aplicación Node.js 24 utilizando una imagen de contenedor extraída de Amazon ECR:

```
$ sam build --use-container --build-image public.ecr.aws/sam/build-nodejs24.x
```

Crea un recurso de función utilizando la imagen del contenedor Python 3.14 extraída de Amazon ECR:

```
$ sam build --use-container --build-image Function1=public.ecr.aws/sam/build-python3.14
```

Implementación gradual de aplicaciones sin servidor con AWS SAM

AWS Serverless Application Model (AWS SAM) viene integrado [CodeDeploy](#) para proporcionar AWS Lambda despliegues graduales. Con solo unas pocas líneas de configuración, AWS SAM hace lo siguiente por usted:

- Implementa nuevas versiones de la función de eventos de Lambda y crea automáticamente alias que apuntan a la nueva versión.
- Cambia gradualmente el tráfico de clientes a la nueva versión hasta que esté satisfecho de que funciona como se esperaba. Si una actualización no funciona correctamente, puedes deshacer los cambios.
- Define funciones de prueba previas y posteriores al tráfico para verificar que el código recién implementado está configurado correctamente y que la aplicación funciona como se espera.
- Revierte automáticamente la implementación si se activan CloudWatch las alarmas.

Note

Si habilita los despliegues graduales a través de su AWS SAM plantilla, se creará automáticamente un CodeDeploy recurso para usted. Puede ver el CodeDeploy recurso directamente a través del Consola de administración de AWS.

Ejemplo

El siguiente ejemplo demuestra el uso de la función Lambda CodeDeploy para trasladar gradualmente a los clientes a la versión recién implementada:

```
Resources:
  MyLambdaFunction:
    Type: AWS::Serverless::Function
    Properties:
      Handler: index.handler
      Runtime: nodejs20.x
      CodeUri: s3://bucket/code.zip

      AutoPublishAlias: live

    DeploymentPreference:
      Type: Canary10Percent10Minutes
      Alarms:
        # A list of alarms that you want to monitor
        - !Ref AliasErrorMetricGreaterThanZeroAlarm
        - !Ref LatestVersionErrorMetricGreaterThanZeroAlarm
      Hooks:
        # Validation Lambda functions that are run before & after traffic shifting
```

```
PreTraffic: !Ref PreTrafficLambdaFunction
PostTraffic: !Ref PostTrafficLambdaFunction
```

Estas revisiones de la AWS SAM plantilla hacen lo siguiente:

- **AutoPublishAlias:** Al añadir esta propiedad y especificar un nombre de alias, AWS SAM:
 - Detecta cuándo se implementa código nuevo, en función de los cambios en el URI de Amazon S3 de la función de Lambda.
 - Crea y publica una versión actualizada de esa función con el código más reciente.
 - Crea un alias con el nombre que usted le indique (a menos que ya exista un alias) y apunta a la versión actualizada de la función de Lambda. Las invocaciones de la función deben utilizar el cualificador de alias para poder usar esta característica. Si no estás familiarizado con el control de versiones y los alias de las funciones de Lambda, consulta [Control de versiones y alias de las funciones de AWS Lambda](#).
- **Deployment Preference Type:** En el ejemplo anterior, el 10 % del tráfico de clientes se desvía inmediatamente a la nueva versión. Transcurridos 10 minutos, todo el tráfico se desvía a la nueva versión. Sin embargo, si las pruebas previas o posteriores al tráfico fallan, o si se activa una CloudWatch alarma, se CodeDeploy revertirá la implementación. Puedes especificar cómo debe desviarse el tráfico entre versiones de las siguientes maneras:
 - **Canary:** el tráfico se desvía en dos incrementos. Puedes elegir entre opciones de valores controlados predefinidos. Las opciones especifican el porcentaje de tráfico que se desvía a la versión actualizada de la función de Lambda en el primer incremento y el intervalo, en minutos, antes de que se desvíe el tráfico restante en el segundo incremento.
 - **Linear:** El tráfico se desvía en incrementos iguales con el mismo número de minutos entre incrementos. Puedes elegir opciones lineales predefinidas que especifiquen el porcentaje de tráfico desviado en cada incremento y el número de minutos entre cada incremento.
 - **AllAtOnce:** todo el tráfico se desvía a la vez desde la función de Lambda original a la versión de la función de Lambda actualizada.

En la siguiente tabla se describen otras opciones de cambio de tráfico que están disponibles además de la utilizada en el ejemplo.

Tipo de preferencia de implementación
Canary10Percent30Minutes

Tipo de preferencia de implementación

Canary10Percent5Minutes

Canary10Percent10Minutes

Canary10Percent15Minutes

Linear10PercentEvery10Minutes

Linear10PercentEvery1Minute

Linear10PercentEvery2Minutes

Linear10PercentEvery3Minutes

AllAtOnce

- **Alarms:** Se trata de CloudWatch alarmas que se activan cuando se produce un error provocado por la implementación. Cuando se detectan, anulan automáticamente la implementación. Por ejemplo, si el código actualizado que está implementando provoca errores en la aplicación. Otro ejemplo es si alguna métrica [AWS Lambda](#) o una CloudWatch métrica personalizada que especificó ha superado el umbral de alarma.
- **Hooks:** estas son funciones de testeo anteriores y posteriores al tráfico que realizan comprobaciones antes de que el tráfico empiece a desviarse a la nueva versión y una vez completado el desvío de tráfico.
 - **PreTraffic:** Antes de que comience el cambio de tráfico, CodeDeploy invoca la función Lambda previa al enlace de tráfico. Esta función Lambda debe volver a llamar CodeDeploy e indicar el éxito o el error. Si la función falla, se interrumpe e informa del error a CloudFormation. Si la función se ejecuta correctamente, CodeDeploy pasa al cambio de tráfico.
 - **PostTraffic:** Una vez finalizado el cambio de tráfico, CodeDeploy invoca la función Lambda posterior al gancho de tráfico. Esto es similar a lo que ocurre antes del bloqueo de tráfico, en el que la función debe volver a llamar CodeDeploy para informar de un éxito o un fracaso. Utiliza enlaces posteriores al tráfico para ejecutar pruebas de integración u otras acciones de validación.

Para obtener más información, consulta [SAM Reference to Safe Deployments](#).

Implementación gradual de una función de eventos de Lambda por primera vez

Cuando se implementa una función Lambda de forma gradual, se CodeDeploy requiere una versión de función previamente implementada desde la que desviar el tráfico. Por lo tanto, su primera implementación debe realizarse en dos pasos:

- Paso 1: implementa la función de Lambda y crea alias automáticamente con AutoPublishAlias.
- Paso 2: haz tu implementación gradual con DeploymentPreference

Al realizar el primer despliegue gradual en dos pasos, se obtiene CodeDeploy una versión anterior de la función Lambda desde la que desplazar el tráfico.

Paso 1: Implemente la función de eventos de Lambda

```
Resources:
MyLambdaFunction:
  Type: AWS::Serverless::Function
  Properties:
    Handler: index.handler
    Runtime: nodejs20.x
    CodeUri: s3://bucket/code.zip

    AutoPublishAlias: live
```

Paso 2: realizar la implementación gradual

```
Resources:
MyLambdaFunction:
  Type: AWS::Serverless::Function
  Properties:
    Handler: index.handler
    Runtime: nodejs20.x
    CodeUri: s3://bucket/code.zip

    AutoPublishAlias: live

  DeploymentPreference:
    Type: Canary10Percent10Minutes
  Alarms:
```



```
# A list of alarms that you want to monitor
- !Ref AliasErrorMetricGreaterThanZeroAlarm
- !Ref LatestVersionErrorMetricGreaterThanZeroAlarm
Hooks:
# Validation Lambda functions that are run before and after traffic shifting
PreTraffic: !Ref PreTrafficLambdaFunction
PostTraffic: !Ref PostTrafficLambdaFunction
```

Más información

Para ver un ejemplo práctico de cómo configurar una implementación gradual, consulta el [Módulo 5: implementaciones de valores controlados](#) en el taller completo de AWS SAM .

Notas de referencia importantes para AWS SAM

Esta sección contiene notas y anuncios importantes para AWS Serverless Application Model (AWS SAM).

Temas

- [Notas importantes para 2023](#)

Notas importantes para 2023

Octubre de 2023

La CLI de AWS SAM deja de dar soporte para Python 3.7

Publicado el 20/10/2023

Python 3.7 recibió end-of-life el estatus en junio de 2023. AWS SAM CLI suspenderá el soporte Python 3.7 el 24 de octubre de 2023. Para obtener más información, consulta el [anuncio](#) en el aws-sam-cli GitHub repositorio.

Este cambio afecta a los siguientes usuarios:

- Si usa Python 3.7 e instala AWS SAM CLI el `pip`.
- Si usas `aws-sam-cli` como biblioteca y crea su aplicación con Python 3.7.

Si lo instala y administra AWS SAM CLI mediante otro método, no se verá afectado.

Para los usuarios afectados, recomendamos que actualicen su entorno de desarrollo a una versión más reciente de Python 3.8 o posterior.

Este cambio no afecta a la compatibilidad con el entorno Python 3.7 AWS Lambda de ejecución.

Para obtener más información, consulta [Política de obsolescencia del tiempo de ejecución](#) en la Guía del desarrollador de AWS Lambda .

AWS SAM CLI Terraform compatibilidad

En esta sección, se describe el uso de la interfaz de línea de AWS Serverless Application Model comandos (AWS SAMCLI) en tus Terraform proyectos y Terraform en la nube.

Para enviar comentarios y solicitudes de características, cree un [Problema de GitHub](#).

Temas

- [Introducción al Terraform soporte para AWS SAM CLI](#)
- [Uso de AWS SAM CLI por Terraform para depuración y pruebas locales](#)
- [Uso de AWS SAM CLI con Serverless.tf para depuración y pruebas locales](#)
- [AWS SAM CLI por Terraform Referencia de](#)
- [¿Qué es? AWS SAM CLI soporte para Terraform?](#)

Introducción al Terraform soporte para AWS SAM CLI

En este tema se explica cómo empezar a utilizar la interfaz de línea de AWS Serverless Application Model comandos (AWS SAMCLI) con Terraform.

Para enviar comentarios y solicitudes de características, cree un [Problema de GitHub](#).

Temas

- [AWS SAM CLI Terraform requisitos previos](#)
- [Utilización AWS SAM CLI comandos con Terraform](#)
- [Configurado para Terraform proyectos](#)
- [Configurado para Terraform Cloud](#)

AWS SAM CLI Terraform requisitos previos

Completa todos los requisitos previos para empezar a utilizar la CLI de AWS SAM en sus proyectos de Terraform.

1. Instala o actualiza la de AWS SAM CLI

Para comprobar si tienes la CLI de AWS SAM instalada, ejecuta lo siguiente:

```
$ sam --version
```

Si la CLI de AWS SAM ya está instalada, el resultado mostrará una versión. Para actualizar a la versión más reciente, consulta [Actualización de AWS SAM CLI](#).

Para obtener instrucciones sobre la instalación de la CLI de AWS SAM junto con todos sus requisitos previos, consulta [Instala la AWS SAM CLI](#).

2. Instalación Terraform

Para comprobar si instalaste Terraform, ejecuta lo siguiente:

```
$ terraform -version
```

Para instalar Terraform, consulta [Instalar Terraform](#) en el registro de Terraform.

3. Instalación Docker para pruebas locales

La CLI de AWS SAM requiere Docker para las pruebas locales. Para instalar Docker, consulta [Instalación de Docker para usarlo con la de AWS SAM CLI](#).

Utilización AWS SAM CLI comandos con Terraform

Cuando ejecuta un comando compatible con la CLI de AWS SAM, utiliza la opción `--hook-name` y proporcione el valor `terraform`. A continuación, se muestra un ejemplo:

```
$ sam local invoke --hook-name terraform
```

Puedes configurar esta opción en su archivo de configuración de la CLI de AWS SAM con lo siguiente:

```
hook_name = "terraform"
```

Configurado para Terraform proyectos

Completa los pasos de este tema para utilizar la CLI de AWS SAM con proyectos de Terraform.

No se requiere ninguna configuración adicional si crea sus AWS Lambda artefactos fuera de su Terraform proyecto. Consulte [Uso de AWS SAM CLI por Terraform para depuración y pruebas locales](#) para empezar a utilizar el AWS SAMCLI.

Si crea los artefactos de Lambda en sus proyectos de Terraform, debes hacer lo siguiente:

1. Instalación de Python 3.8 o posterior
2. Instala la herramienta Make.
3. Define la lógica de creación de artefactos de Lambda en su proyecto de Terraform.
4. Define un recurso `sam metadata` para informar su lógica de compilación a la CLI de AWS SAM.
5. Utilice el AWS SAMCLI `sam build` comando para crear sus artefactos Lambda.

Instalación Python 3.8 o más reciente

PythonSe requiere 3.8 o una versión más reciente para su uso con AWS SAMCLI. Cuando se ejecuta `sam build`, la AWS SAM de Python crea `makefiles` que contiene comandos de CLI para crear sus artefactos de Lambda.

Para obtener instrucciones de instalación, consulta la página [Downloading Python](#) (Descarga de Python) en la Guía para principiantes de Python.

Compruebe que Python 3.8 o una versión posterior esté agregada a la ruta de su máquina ejecutando:

```
$ python --version
```

El resultado debe mostrar una versión de Python 3.8 o posterior.

Instala la Make herramienta

GNU[Make](#) es una herramienta que controla la generación de ejecutables y otros archivos que no son el origen del proyecto. La CLI de AWS SAM crea `makefiles` que se basa en esta herramienta para crear sus artefactos de Lambda.

Si no has instalado Make en su máquina local, instálalo antes de continuar.

Para Windows, puedes realizar la instalación mediante [Chocolatey](#). Para obtener instrucciones, consulta [Uso de Chocolatey](#) en Cómo instalar y usar “Make” en Windows

Define la lógica de construcción de los artefactos de Lambda

Utiliza el tipo de recurso `null_resource` Terraform para definir la lógica de compilación de Lambda. El siguiente es un ejemplo que utiliza un script de compilación personalizado para crear una función de eventos Lambda.

```
resource "null_resource" "build_lambda_function" {
  triggers = {
    build_number = "${timestamp()}"
  }

  provisioner "local-exec" {
    command = substr(pathexpand("~"), 0, 1) == "/" ? "./" : ""
    py_build.sh "${local.lambda_src_path}" "${local.building_path}"
    "${local.lambda_code_filename}" Function : "powershell.exe -File .\\PyBuild.ps1
    ${local.lambda_src_path} ${local.building_path} ${local.lambda_code_filename}
    Function"
  }
}
```

Defina un metadatos de sam recurso

El recurso `sam_metadata` es un tipo de recurso `null_resource` Terraform que proporciona a la CLI de AWS SAM la información que necesita para localizar los artefactos de Lambda. Se requiere un recurso de `sam_metadata` único para cada función o capa de Lambda del proyecto. Para obtener más información sobre este tipo de recurso, consulta [null_resource](#) en el registro de Terraform.

Para definir un metadatos de sam recurso

1. Asigne un nombre al recurso empezando por `sam_metadata_` para identificar el recurso como un recurso `sam_metadata`.
2. Defina las propiedades de sus artefactos Lambda dentro del bloque de `triggers` de tu recurso.
3. Especifica el `null_resource` que contiene la lógica de compilación de Lambda con el argumento `depends_on`.

La siguiente es una plantilla de ejemplo:

```
resource "null_resource" "sam_metadata_..." {
  triggers = {
```

```

    resource_name = resource_name
    resource_type = resource_type
    original_source_code = original_source_code
    built_output_path = built_output_path
  }
  depends_on = [
    null_resource.build_lambda_function # ref to your build logic
  ]
}
```

El siguiente es un recurso sam metadata de ejemplo:

```

resource "null_resource" "sam_metadata_aws_lambda_function_publish_book_review" {
  triggers = {
    resource_name = "aws_lambda_function.publish_book_review"
    resource_type = "ZIP_LAMBDA_FUNCTION"
    original_source_code = "${local.lambda_src_path}"
    built_output_path = "${local.building_path}/${local.lambda_code_filename}"
  }
  depends_on = [
    null_resource.build_lambda_function
  ]
}
```

El contenido del recurso sam metadata variará según el tipo de recurso de Lambda (función o capa) y el tipo de empaquetado (ZIP o imagen). Para obtener más información y ejemplos, consulta [recurso de metadatos de sam](#).

Al configurar un recurso sam metadata y utilizar un comando compatible con la CLI de AWS SAM, la CLI de AWS SAM generará el archivo de metadatos antes de ejecutar el comando de la CLI de AWS SAM. Una vez hayas generado este archivo, puedes usar la opción `--skip-prepare-infra` con comandos futuros de la CLI de AWS SAM para omitir el proceso de generación de metadatos y ahorrar tiempo. Esta opción solo debe usarse si no ha realizado ningún cambio en la infraestructura, como la creación de nuevas funciones de Lambda o nuevos puntos de conexión de API.

Use la AWS SAM CLI para crear sus artefactos Lambda

Utilice el AWS SAMCLI `sam build` comando para crear sus artefactos Lambda. Cuando se ejecuta `sam build`, la CLI de AWS SAM hace lo siguiente:

1. Busca recursos `aws_lambda_function` en tu proyecto de Terraform para conocer y localizar tus recursos de Lambda.
2. Inicia la lógica de compilación de Lambda para crear sus artefactos de Lambda.
3. Crea un `.aws-sam` directorio que organiza el Terraform proyecto para usarlo con los AWS SAMCLI `aws-sam local` comandos.

Para crear con `aws-sam build`

1. Desde el directorio que contiene el módulo raíz de Terraform, ejecuta lo siguiente:

```
$ aws-sam build --hook-name terraform
```

2. Para crear una función o capa de Lambda específica, ejecuta lo siguiente

```
$ aws-sam build --hook-name terraform lambda-resource-id
```

El ID de recurso de Lambda puede ser el nombre de la función de Lambda o la dirección completa del recurso de Terraform, como `aws_lambda_function.list_books` o `module.list_book_function.aws_lambda_function.this[0]`.

Si el código de origen de la función u otros archivos de configuración de Terraform se encuentran fuera del directorio que contiene el módulo raíz de Terraform, debes especificar la ubicación. Utiliza la opción `--terraform-project-root-path` para especificar la ruta absoluta o relativa al directorio de nivel superior que contiene estos archivos. A continuación, se muestra un ejemplo:

```
$ aws-sam build --hook-name terraform --terraform-project-root-path ~/projects/terraform/demo
```

Creación mediante un contenedor

Al ejecutar el AWS SAMCLI `aws-sam build` comando, puede configurarlo AWS SAMCLI para que compile su aplicación mediante un Docker contenedor local.

Note

Debe tener Docker instalado y configurado. Para obtener instrucciones, consulta [Instalación de Docker para usarlo con la de AWS SAM CLI](#).

Para compilar usando un contenedor

1. Crea un Dockerfile que contenga las herramientas Terraform, Python y Make. También debes incluir el tiempo de ejecución de la función de Lambda.

A continuación, se muestra un Dockerfile de ejemplo:

```
FROM public.ecr.aws/amazonlinux/amazonlinux:2

RUN yum -y update \
    && yum install -y unzip tar gzip bzip2-devel ed gcc gcc-c++ gcc-gfortran \
    less libcurl-devel openssl openssl-devel readline-devel xz-devel \
    zlib-devel glibc-static libgcc libgcc-devel llvm-toolset-7 zlib-static \
    && rm -rf /var/cache/yum

RUN yum -y install make \
    && yum -y install zip

RUN yum install -y yum-utils \
    && yum-config-manager --add-repo https://rpm.releases.hashicorp.com/
AmazonLinux/hashicorp.repo \
    && yum -y install terraform \
    && terraform --version

# AWS Lambda Builders
RUN amazon-linux-extras enable python3.8
RUN yum clean metadata && yum -y install python3.8
RUN curl -L get-pip.io | python3.8
RUN pip3 install aws-lambda-builders
RUN ln -s /usr/bin/python3.8 /usr/bin/python3
RUN python3 --version

VOLUME /project
WORKDIR /project

ENTRYPOINT ["sh"]
```

2. Use [docker build](#) para crear su imagen de Docker.

A continuación, se muestra un ejemplo:

```
$ docker build --tag terraform-build:v1 <path-to-directory-containing-Dockerfile>
```

3. Ejecute el AWS SAMCLI `sam build` comando con las `--build-image` opciones `--use-container` y.

A continuación, se muestra un ejemplo:

```
$ sam build --use-container --build-image terraform-build:v1
```

Pasos a seguir a continuación

Para empezar a utilizar la CLI de AWS SAM en sus proyectos de Terraform, consulta [Uso de AWS SAM CLI por Terraform para depuración y pruebas locales](#).

Configurado para Terraform Cloud

Se recomienda usar Terraform v1.6.0 o posterior. Si usa una versión anterior, debes generar un archivo de plan de Terraform de forma local. El archivo del plan local AWS SAM CLI proporciona la información que necesita para realizar las pruebas y la depuración locales.

Para generar un archivo de plan local

Note

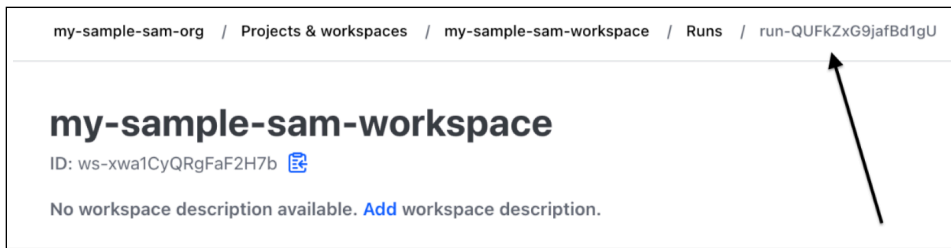
Estos pasos no son necesarios con Terraform v1.6.0 o posterior. Para empezar a utilizar el AWS SAM CLI with Terraform Cloud, consulte [Uso de la CLI de AWS SAM con Terraform](#).

1. Configurar un token de API: el tipo de token dependerá de su nivel de acceso. Para obtener más información, consulta [Tokens de API](#) en la documentación de Terraform Cloud.
2. Configurar la variable de entorno del token de la API: el siguiente es un ejemplo de la línea de comandos:

```
$ export TOKEN="<api-token-value>"
```

3. Obtenga su ID de ejecución: en la Terraform Cloud consola, busque el ID de Terraform ejecución de la ejecución que desee usar con el AWS SAMCLI.

El ID de ejecución se encuentra en la ruta de navegación del proceso de ejecución.



4. Obtener el archivo del plan: con su token de API, obtengas el archivo del plan local. A continuación, se muestra un ejemplo de la línea de comando:

```
curl \
  --header "Authorization: Bearer $TOKEN" \
  --header "Content-Type: application/vnd.api+json" \
  --location \
  https://app.terraform.io/api/v2/runs/<run ID>/plan/json-output \
  > custom_plan.json
```

Ahora ya puedes utilizar la CLI de AWS SAM con Terraform Cloud. Si utilizas un comando compatible con la CLI de AWS SAM, utiliza la opción `--terraform-plan-file` para especificar el nombre y la ruta del archivo del plan local. A continuación, se muestra un ejemplo:

```
$ sam local invoke --hook-name terraform --terraform-plan-file custom-plan.json
```

A continuación, se muestra un ejemplo con el comando `sam local start-api`:

```
$ sam local start-api --hook-name terraform --terraform-plan-file custom-plan.json
```

Para ver una aplicación de muestra que pueda usar con estos ejemplos, consulta [api_gateway_v2_tf_cloud](#) en el repositorio de GitHub `aws-samples`.

Pasos a seguir a continuación

Para comenzar a usar la CLI de AWS SAM con Terraform Cloud, consulta [Uso de AWS SAM CLI por Terraform para depuración y pruebas locales](#).

Uso de AWS SAM CLI por Terraform para depuración y pruebas locales

En este tema se explica cómo utilizar los AWS Serverless Application Model comandos de la interfaz de línea de comandos (AWS SAMCLI) compatibles con sus Terraform proyectos y Terraform Cloud.

Para enviar comentarios y solicitudes de características, cree un [Problema de GitHub](#).

Temas

- [Pruebas locales con la de invocar sam local](#)
- [Pruebas locales con la de sam local start-api](#)
- [Pruebas locales con la de sam local start-lambda](#)
- [Terraform limitaciones](#)

Pruebas locales con la de invocar sam local

Note

Para usar la CLI de AWS SAM para realizar pruebas locales, debe tener Docker instalado y configurado. Para obtener instrucciones, consulte [Instalación de Docker para usarlo con la de AWS SAM CLI](#).

A continuación, se muestra un ejemplo de cómo probar la función de eventos de Lambda de forma local mediante la transmisión de un evento:

```
$ sam local invoke --hook-name terraform hello_world_function -e events/event.json -
```

Para obtener más información acerca de este comando, consulta [Introducción a las pruebas con invocar sam local](#).

Pruebas locales con la de sam local start-api

Para utilizar `sam local start-api` con Terraform, ejecuta lo siguiente:

```
$ sam local start-api --hook-name terraform
```

A continuación, se muestra un ejemplo:

```
$ sam local start-api --hook-name terraform
```

Running Prepare Hook to prepare the current application

Executing prepare hook of hook "terraform"

Initializing Terraform application

...
Creating terraform plan and getting JSON output

....
Generating metadata file

Unresolvable attributes discovered in project, run terraform apply to resolve them.

Finished generating metadata file. Storing in...
Prepare hook completed and metadata file generated at: ...
Mounting HelloWorldFunction at http://127.0.0.1:3000/hello [GET]

Mounting None at http://127.0.0.1:3000/hello [POST]

```
You can now browse to the above endpoints to invoke your functions. You do not need
to restart/reload SAM CLI while working on your functions, changes will be reflected
instantly/automatically. If you
used sam build before running local commands, you will need to re-run sam build for the
changes to be picked up. You only need to restart SAM CLI if you update your AWS SAM
template
2023-06-26 13:21:20 * Running on http://127.0.0.1:3000/ (Press CTRL+C to quit)
```

Para obtener más información acerca de este comando, consulta [Introducción a las pruebas con sam local start-api](#).

Funciones de eventos de Lambda que utilizan autorizadores de Lambda

En el caso de las funciones de Lambda configuradas para utilizar autorizadores de Lambda, la CLI de AWS SAM invocará automáticamente el autorizador de Lambda antes de invocar el punto de conexión de la función de Lambda.

- Para obtener más información sobre esta función en AWS SAMCLI, consulte [Funciones de eventos de Lambda que utilizan autorizadores de Lambda](#).
- Para obtener más información sobre el uso de autorizadores Lambda en Terraform, consulta [Resource: aws_api_gateway_authorizer](#) en el registro de Terraform.

Pruebas locales con la de sam local start-lambda

El siguiente es un ejemplo de cómo probar la función Lambda localmente con AWS Command Line Interface (AWS CLI):

1. Utiliza la CLI de AWS SAM para crear un entorno de testeo local:

```
$ sam local start-lambda --hook-name terraform hello_world_function
```

2. Utilice el AWS CLI para invocar la función de forma local:

```
$ aws lambda invoke --function-name hello_world_function --endpoint-
url http://127.0.0.1:3001/ response.json --cli-binary-format raw-in-base64-out --
payload file://events/event.json
```

Para obtener más información acerca de este comando, consulta [Introducción a las pruebas con sam local start-lambda](#).

Terraform limitaciones

Las siguientes son limitaciones cuando se usa la CLI de AWS SAM con Terraform:

- Funciones de Lambda vinculadas a varias capas.
- Variables locales de Terraform que definen los vínculos entre los recursos.
- Referencia a una función de Lambda que aún no se ha creado. Esto incluye las funciones que se definen en el atributo `body` del recurso de la API de REST.

Para evitar estas limitaciones, puedes ejecutar `terraform apply` cuando se agrega un nuevo recurso.

Uso de AWS SAM CLI con Serverless.tf para depuración y pruebas locales

La interfaz de línea de AWS Serverless Application Model comandos (AWS SAMCLI) se puede utilizar con Serverless.tf módulos para la depuración y las pruebas locales de las funciones y capas de AWS Lambda eventos. Se admiten los siguientes comandos de la CLI de AWS SAM:

- `sam build`
- `sam local invoke`
- `sam local start-api`
- `sam local start-lambda`

Note

Serverless.tf la versión 4.6.0 y posteriores admiten AWS SAMCLI la integración.

Para empezar a utilizarla AWS SAMCLI con sus Serverless.tf módulos, actualice a la versión más reciente de Serverless.tf y a. AWS SAMCLI

A partir de la versión 6.0.0 de serverless.tf, debes establecer el parámetro `create_sam_metadata` como `true`. Esto genera los recursos de metadatos que requiere el AWS SAMCLI `sam build` comando.

Para obtener más información sobre Serverless.tf, consulta el módulo [terraform-aws-lambda-module](#).

AWS SAM CLI por Terraform Referencia de

Esta sección es la referencia para usar la interfaz de línea de AWS Serverless Application Model comandos (AWS SAMCLI) con Terraform fines de depuración y pruebas locales.

Para enviar comentarios y solicitudes de características, cree un [Problema de GitHub](#).

AWS SAM referencia de funciones compatibles

La documentación de referencia sobre las características de la CLI de AWS SAM que se admiten para su uso con Terraform se encuentra aquí:

- [compilación de sam](#)
- [invocar sam local](#)
- [sam local start-api](#)
- [sam local start-lambda](#)

Terraform referencia específica

La documentación de referencia específica para usar la CLI de AWS SAM con Terraform se encuentra aquí:

- [recurso de metadatos de sam](#)

recurso de metadatos de sam

Esta página contiene información de referencia sobre el tipo de recurso sam metadata resource utilizado en los proyectos de Terraform.

- Para obtener una introducción al uso de la interfaz de línea de AWS Serverless Application Model comandos (AWS SAMCLI) con Terraform, consulte [¿Qué es? AWS SAM CLI soporte para Terraform?](#).
- Para utilizar la CLI de AWS SAM con Terraform, consulta [Uso de AWS SAM CLI por Terraform para depuración y pruebas locales](#).

Temas

- [Argumentos](#)
- [Ejemplos](#)

Argumentos

Argumento	Description (Descripción)
<code>built_output_path</code>	El camino hacia los artefactos creados por tu función de AWS Lambda eventos.
<code>docker_build_args</code>	Cadena decodificada del objeto JSON de argumentos de compilación de Docker. Este argumento es opcional.
<code>docker_context</code>	La ruta al directorio que contiene el contexto de creación de la imagen de Docker.
<code>docker_file</code>	<p>La ruta del archivo de Docker. Esta ruta es relativa a la ruta <code>docker_context</code> .</p> <p>Este argumento es opcional. El valor predeterminado es <code>Dockerfile</code> .</p>
<code>docker_tag</code>	El valor de la etiqueta de imagen de Docker creada. Este valor es opcional.
<code>depends_on</code>	La ruta al recurso de construcción para la función o capa de Lambda. Para obtener más información, consulta El argumento depends_on en el registro de Terraform.
<code>original_source_code</code>	<p>La ruta a la que se define la función de Lambda. Este valor puede ser una cadena, una matriz de cadenas o un objeto JSON decodificado como cadena.</p> <ul style="list-style-type: none">• En el caso de las matrices de cadenas, solo se utiliza el primer valor, ya que no se admiten varias rutas de código.• En el caso de los objetos JSON, también debes definir la <code>source_code_property</code> .

Argumento	Description (Descripción)
resource_name	El nombre de la función de Lambda.
resource_type	<p>El formato del tipo de paquete de funciones de Lambda. Los valores aceptados son:</p> <ul style="list-style-type: none">• IMAGE_LAMBDA_FUNCTION• LAMBDA_LAYER• ZIP_LAMBDA_FUNCTION
source_code_property	La ruta al código de recursos de Lambda en el objeto JSON. Define esta propiedad cuando original_source_code sea un objeto JSON.

Ejemplos

recurso de metadatos sam que hace referencia a una función de eventos de Lambda mediante el tipo de paquete ZIP

```
# Lambda function resource
resource "aws_lambda_function" "tf_lambda_func" {
  filename = "${path.module}/python/hello-world.zip"
  handler  = "index.lambda_handler"
  runtime  = "python3.8"
  function_name = "function_example"
  role      = aws_iam_role.iam_for_lambda.arn
  depends_on = [
    null_resource.build_lambda_function # function build logic
  ]
}

# sam metadata resource
resource "null_resource" "sam_metadata_function_example" {
  triggers = {
    resource_name = "aws_lambda_function.function_example"
    resource_type = "ZIP_LAMBDA_FUNCTION"
    original_source_code = "${path.module}/python"
    built_output_path = "${path.module}/building/function_example"
  }
}
```

```
depends_on = [  
    null_resource.build_lambda_function # function build logic  
]  
}
```

sam: recurso de metadatos que hace referencia a una función de eventos de Lambda mediante el tipo de paquete de imágenes

```
resource "null_resource" "sam_metadata_function" {  
  triggers = {  
    resource_name = "aws_lambda_function.image_function"  
    resource_type = "IMAGE_LAMBDA_FUNCTION"  
    docker_context = local.lambda_src_path  
    docker_file = "Dockerfile"  
    docker_build_args = jsonencode(var.build_args)  
    docker_tag = "latest"  
  }  
}
```

recurso de metadatos sam que hace referencia a una capa de Lambda

```
resource "null_resource" "sam_metadata_layer1" {  
  triggers = {  
    resource_name = "aws_lambda_layer_version.layer"  
    resource_type = "LAMBDA_LAYER"  
    original_source_code = local.layer_src  
    built_output_path = "${path.module}/${layer_build_path}"  
  }  
  depends_on = [null_resource.layer_build]  
}
```

¿Qué es? AWS SAM CLI soporte para Terraform?

Utilice la interfaz de línea de AWS Serverless Application Model comandos (AWS SAMCLI) con sus Terraform proyectos o Terraform Cloud para realizar depuraciones y pruebas locales de:

- AWS Lambda Funciones y capas de eventos.
- API de Amazon API Gateway.

Para ver una introducción a Terraform, consulta [¿Qué es Terraform?](#) en el sitio web HashiCorp Terraform.

Para enviar comentarios y solicitudes de características, cree un [Problema de GitHub](#).

Note

Como parte del paso de análisis de la integración, AWS SAMCLI los comandos AWS SAMCLI del usuario de los procesos generan archivos y datos del proyecto. El resultado del comando debe permanecer sin cambios, pero en algunos entornos, el entorno o el ejecutor pueden introducir registros o información adicionales en el resultado.

Temas

- [¿Qué es el AWS SAM CLI?](#)
- [¿Cómo utilizo el AWS SAM CLI por Terraform?](#)
- [Pasos a seguir a continuación](#)

¿Qué es el AWS SAM CLI?

AWS SAMCLIEs una herramienta de línea de comandos que puede utilizar con AWS SAM plantillas e integraciones de terceros compatibles, por ejemploTerraform, para crear y ejecutar aplicaciones sin servidor. Para obtener una introducción a la AWS SAMCLI, consulte. [¿Qué es el AWS SAM CLI?](#)

The AWS SAMCLI admite los siguientes comandos paraTerraform:

- `sam local invoke`— Inicie una única invocación local de un recurso de AWS Lambda función. Para obtener más información acerca de este comando, consulte [Introducción a las pruebas con invocar sam local](#).
- `sam local start-api`: ejecuta tus recursos de Lambda localmente y pruébalos a través de un host de servidor HTTP local. Este tipo de pruebas resulta útil para las funciones de Lambda que invoca un punto de conexión de API Gateway. Para obtener más información acerca de este comando, consulte [Introducción a las pruebas con sam local start-api](#).
- `sam local start-lambda`— Inicie un punto final local para su función Lambda a fin de invocarla localmente mediante AWS Command Line Interface (AWS CLI) o SDK. Para obtener más información acerca de este comando, consulte [Introducción a las pruebas con sam local start-lambda](#).

¿Cómo utilizo el AWS SAM CLI por Terraform?

El [flujo de trabajo principal de Terraform](#) consta de tres etapas: escribir, planificar y aplicar. Con la AWS SAMCLI compatibilidad conTerraform, puede aprovechar el AWS SAMCLI `sam local` conjunto de comandos y, al mismo tiempo, seguir utilizando sus Terraform flujos de trabajo para administrar sus aplicaciones AWS. En general, esto significa lo siguiente:

- Escribir: cree su infraestructura como código mediante Terraform.
- Probar y depurar: usa la CLI de AWS SAM para probar y depurar sus aplicaciones de forma local.
- Planificar: obtengas una vista previa de los cambios antes de aplicarlos.
- Aplicar: aprovisiona su infraestructura.

Para ver un ejemplo del uso de AWS SAMCLI withTerraform, consulta [Better together: AWS SAMCLI y HashiCorp Terraform](#) en el blog de AWS informática.

Pasos a seguir a continuación

Para completar todos los requisitos previos y configurar Terraform, consulta [Introducción al Terraform soporte para AWS SAM CLI](#).

Publicación de la aplicación con la de AWS SAM CLI

Para que su AWS SAM aplicación esté disponible para que otros la encuentren y la desplieguen, puede utilizarla AWS SAMCLI para publicarla en AWS Serverless Application Repository. Para publicar su aplicación mediante el AWS SAMCLI, debe definirla mediante una AWS SAM plantilla. También debes haberla testeado localmente o en AWS la nube.

Sigue las instrucciones de este tema para crear una nueva aplicación, crear una nueva versión de una aplicación existente o actualizar los metadatos de una aplicación existente. (Lo que haga dependerá de si la aplicación ya existe en la AWS Serverless Application Repository aplicación y de si algún metadato de la aplicación está cambiando). Para obtener más información acerca de los metadatos de la aplicación, consulta [AWS SAM propiedades de la sección de metadatos de la plantilla](#).

Requisitos previos

Antes de publicar una aplicación en el AWS Serverless Application Repository mediante AWS SAMCLI, debe disponer de lo siguiente:

- La (CLI) de AWS SAM debe estar instalada Para obtener más información, consulta [Instala la AWS SAM CLI](#). Para determinar si la CLI de AWS SAM está instalado, ejecuta el siguiente comando:

```
sam --version
```

- Una AWS SAM plantilla válida.
- El código de su aplicación y las dependencias a las que hace referencia la AWS SAM plantilla.
- Una versión semántica, necesaria solo para compartir la aplicación públicamente. Este valor puedes ser tan simple como 1.0.
- Una URL que apunta al código fuente de la aplicación.
- Un archivo README.md. En este archivo se describe cómo se utiliza la aplicación y cómo configurarla antes de implementarla en las cuentas de AWS .
- Un archivo LICENSE.txt, necesario solo para compartir la aplicación públicamente.
- Si tu aplicación contiene aplicaciones anidadas, debe haberlas publicado ya en AWS Serverless Application Repository.

- Una política de bucket válida de Amazon Simple Storage Service (Amazon S3) que conceda al servicio permisos de lectura para los artefactos que cargue en Amazon S3 cuando empaquete tu aplicación. Para configurar esta política, haga lo siguiente:
 1. Abra la consola de Amazon S3 en <https://console.aws.amazon.com/s3/>.
 2. Elige el nombre del bucket de Amazon S3 que utilizaste para empaquetar tu aplicación.
 3. Elige Permisos.
 4. En la pestaña Permissions (Permisos), en Bucket policy (Política de bucket), elige Edit (Editar).
 5. En la página Editar política de bucket, pegue la siguiente instrucción de Política en el editor de políticas. En la instrucción de política, asegúrese de usar el nombre del bucket en el elemento Resource y el ID de la cuenta AWS en el elemento Condition. La expresión del Condition elemento garantiza que AWS Serverless Application Repository tiene permiso para acceder únicamente a las aplicaciones de la AWS cuenta especificada. Para obtener más información acerca de las secciones dentro de una instrucción de política de IAM, consulta [Referencia de los elementos de la política de IAM](#) en la Guía del usuario de IAM.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "serverlessrepo.amazonaws.com"
      },
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::<your-bucket-name>/*",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        }
      }
    }
  ]
}
```

6. Elige Guardar cambios.

Publicación de una aplicación

Paso 1: Añada una sección de **metadatos** a AWS SAM plantilla

Primero, agrega una Metadata sección a tu AWS SAM plantilla. Proporcione la información de la solicitud que se publicará en AWS Serverless Application Repository.

A continuación, se ofrece un ejemplo de sección Metadata:

```
Metadata:
  AWS::ServerlessRepo::Application:
    Name: my-app
    Description: hello world
    Author: user1
    SpdxLicenseId: Apache-2.0
    LicenseUrl: LICENSE.txt
    ReadmeUrl: README.md
    Labels: ['tests']
    HomePageUrl: https://github.com/user1/my-app-project
    SemanticVersion: 0.0.1
    SourceCodeUrl: https://github.com/user1/my-app-project

Resources:
  HelloWorldFunction:
    Type: AWS::Lambda::Function
    Properties:
      ...
      CodeUri: source-code1
      ...
```

Para obtener más información sobre la Metadata sección de la AWS SAM plantilla, consulte [AWS SAM propiedades de la sección de metadatos de la plantilla](#).

Paso 2: crear el paquete de la aplicación

Ejecuta el siguiente comando en la CLI de AWS SAM, que carga los artefactos de la aplicación en Amazon S3 y genera un nuevo archivo de plantilla llamado `packaged.yaml`:

```
sam package --output-template-file packaged.yaml --s3-bucket <your-bucket-name>
```


En el siguiente paso, utiliza el archivo de plantilla `packaged.yaml` para publicar la aplicación en AWS Serverless Application Repository. Este archivo es similar al archivo de plantilla original (`template.yaml`), pero tiene una diferencia clave: las propiedades `CodeUri`, `LicenseUrl`, y `ReadmeUrl` apuntan al bucket de Amazon S3 y a los objetos que contienen los artefactos respectivos.

El siguiente fragmento de un ejemplo de archivo de plantilla `packaged.yaml` muestra la propiedad `CodeUri`:

```
MySampleFunction:
  Type: AWS::Serverless::Function
  Properties:
    CodeUri: s3://bucketname/fbd77a3647a4f47a352fc0bjectGUID

...
```

Paso 3: publicar la aplicación

Para publicar una versión privada de la AWS SAM aplicación en AWS Serverless Application Repository, ejecute el siguiente AWS SAMCLI comando:

```
sam publish --template packaged.yaml --region us-east-1
```

El resultado del comando `sam publish` incluye un enlace a tu aplicación en el AWS Serverless Application Repository. También puedes ir directamente a la [AWS Serverless Application Repository página de inicio](#) y buscar tu solicitud.

Paso 4: compartir la aplicación (opcional)

De forma predeterminada, la aplicación está configurada como privada, por lo que no es visible para otras cuentas de AWS. Para compartir tu aplicación con otras personas, debes hacerla pública o conceder permiso a una lista específica de AWS cuentas.

Para obtener información sobre cómo compartir tu aplicación mediante el AWS CLI, consulta los [ejemplos AWS Serverless Application Repository Resource-Based de políticas](#) en la Guía para AWS Serverless Application Repository desarrolladores. Para obtener información sobre cómo compartir tu aplicación mediante la Consola de administración de AWS, consulta [Compartir una aplicación](#) en la Guía para desarrolladores de AWS Serverless Application Repository.

Publicar una nueva versión de una aplicación existente

Después de publicar una aplicación en el AWS Serverless Application Repository, es posible que desee publicar una nueva versión de la misma. Por ejemplo, es posible que haya cambiado el código de la función de eventos Lambda o que haya agregado un nuevo componente a la arquitectura de la aplicación.

Para actualizar una aplicación que hayas publicado anteriormente, vuelves a publicarla siguiendo el mismo proceso detallado anteriormente. En la sección Metadata del archivo de plantilla de AWS SAM, proporciona el mismo nombre de aplicación con el que la publicó originalmente, pero incluya un valor `SemanticVersion` nuevo.

Por ejemplo, considera una aplicación publicada con el nombre `SampleApp` y una `SemanticVersion` de `1.0.0`. Para actualizar esa aplicación, la plantilla de AWS SAM debe tener el nombre de la aplicación `SampleApp` y una `SemanticVersion` de `1.0.1` (o cualquier otro nombre que no sea `1.0.0`).

Temas adicionales

- [AWS SAM propiedades de la sección de metadatos de la plantilla](#)

AWS SAM propiedades de la sección de metadatos de la plantilla

`AWS::ServerlessRepo::Application` es una clave de metadatos que puede usar para especificar la información de la aplicación que desea que se publique en AWS Serverless Application Repository.

Note

CloudFormation la clave de `AWS::ServerlessRepo::Application` metadatos no admite las [funciones intrínsecas](#).

Propiedades

En esta tabla se proporciona información sobre las propiedades de la Metadata sección de la AWS SAM plantilla. Esta sección es necesaria para publicar aplicaciones AWS Serverless Application Repository en el AWS SAMCLI.

Propiedad	Tipo	Obligatorio	Descripción
Name	Cadena	TRUE	<p>Nombre de la aplicación.</p> <p>Longitud mínima = 1. Longitud máxima = 140.</p> <p>Patrón: "[a-zA-Z0-9\\-]+";</p>
Description	Cadena	TRUE	<p>La descripción de la aplicación.</p> <p>Longitud mínima = 1. Longitud máxima = 256.</p>
Author	Cadena	TRUE	<p>El nombre del autor que publica la aplicación.</p> <p>Longitud mínima = 1. Longitud máxima = 127.</p> <p>Patrón: "^([a-z0-9]([a-z0-9] -(?!-))*[a-z0-9])?\$";</p>
SpdxLicenseId	Cadena	FALSO	<p>Un identificador de licencia válido. Para ver la lista de identificadores de licencia válidos, consulta la Lista de licencias de SPDX en el sitio web Package Data Exchange (SPDX).</p>
LicenseUrl	Cadena	FALSO	<p>La referencia a un archivo de licencia local, o un enlace de Amazon S3 a un archivo de licencia, que coincide con el valor SPDXLicenseId de su aplicación.</p> <p>Un archivo de AWS SAM plantilla que no se haya empaquetado con el <code>sam package</code> comando puede tener una referencia a un archivo local para esta propiedad. Sin embargo, para que una aplicación se publique mediante el comando <code>sam publish</code>, esta propiedad debe hacer referencia a un bucket de Amazon S3.</p> <p>Tamaño máximo: 5 MB.</p>

Propiedad	Tipo	Obligatorio	Descripción
			Debe proporcionar un valor de esta propiedad para que la aplicación sea pública. Tenga en cuenta que no puede actualizar esta propiedad una vez publicada la aplicación. Por lo tanto, para añadir una licencia a una aplicación, primero debe eliminarla o publicar una nueva aplicación con un nombre diferente.
ReadmeUrl	Cadena	FALSO	<p>La referencia a un archivo readme local o un enlace de Amazon S3 al archivo readme que contiene una descripción más detallada de la aplicación y de su funcionamiento.</p> <p>Un archivo de AWS SAM plantilla que no se haya empaquetado con el <code>sam package</code> comando puede tener una referencia a un archivo local para esta propiedad. Sin embargo, para que se publique mediante el comando <code>sam publish</code>, esta propiedad debe ser una referencia a un bucket de Amazon S3.</p> <p>Tamaño máximo: 5 MB.</p>
Labels	Cadena	FALSO	<p>Las etiquetas que mejoran la detección de las aplicaciones en los resultados de búsqueda.</p> <p>Longitud mínima = 1. Longitud máxima = 127. Número máximo de etiquetas: 10.</p> <p>Patrón: <code>^[a-zA-Z0-9+\\-_\\:\\@]+\$</code>;</p>
HomePageUrl	Cadena	FALSO	Una URL con más información sobre la aplicación, por ejemplo, la ubicación del GitHub repositorio de la aplicación.

Propiedad	Tipo	Obligatorio	Descripción
SemanticVersion	Cadena	FALSO	<p>La versión semántica de la aplicación. Para obtener más información sobre el control de versiones semántico, consulta el sitio web Control de versiones semántico.</p> <p>Debe proporcionar un valor de esta propiedad para que la aplicación sea pública.</p>
SourceCodeUrl	Cadena	FALSO	Un enlace a un repositorio público para el código fuente de la aplicación.

Casos de uso

En esta sección se enumeran los casos de uso para publicar aplicaciones, junto con las propiedades Metadata que se procesan para ese caso de uso. Se omiten las propiedades que no aparecen en la lista para un caso de uso determinado.

- Creación de una nueva aplicación: se crea una nueva aplicación si no hay ninguna aplicación AWS Serverless Application Repository con el mismo nombre para una cuenta.
 - Name
 - SpdxLicenseId
 - LicenseUrl
 - Description
 - Author
 - ReadmeUrl
 - Labels
 - HomePageUrl
 - SourceCodeUrl
 - SemanticVersion
 - El contenido de la AWS SAM plantilla (por ejemplo, cualquier fuente de eventos, recursos y código de la Función de Evento Lambda)

- Creación de una versión de la aplicación: se crea una versión de la aplicación si ya existe una aplicación AWS Serverless Application Repository con el mismo nombre para una cuenta y SemanticVersion está cambiando.
 - Description
 - Author
 - ReadmeUrl
 - Labels
 - HomePageUrl
 - SourceCodeUrl
 - SemanticVersion
 - El contenido de la AWS SAM plantilla (por ejemplo, cualquier fuente de eventos, recursos y código de función Lambda)
- Actualización de una aplicación: una aplicación se actualiza si ya existe una aplicación AWS Serverless Application Repository con el mismo nombre para una cuenta y SemanticVersion no cambia.
 - Description
 - Author
 - ReadmeUrl
 - Labels
 - HomePageUrl

Ejemplo

A continuación, se ofrece un ejemplo de sección Metadata:

```
Metadata:
  AWS::ServerlessRepo::Application:
    Name: my-app
    Description: hello world
    Author: user1
    SpdxLicenseId: Apache-2.0
```

```
LicenseUrl: LICENSE.txt  
ReadmeUrl: README.md  
Labels: ['tests']  
HomePageUrl: https://github.com/user1/my-app-project  
SemanticVersion: 0.0.1  
SourceCodeUrl: https://github.com/user1/my-app-project
```

Historial de documentos para AWS SAM

En la siguiente tabla se describen los cambios importantes en cada versión de la Guía para desarrolladores de AWS Serverless Application Model . Para obtener notificaciones sobre las actualizaciones de esta documentación, puedes suscribirte a una fuente RSS.

- Última actualización de la documentación: 9 de octubre de 2026

Cambio	Descripción	Fecha
Se agregó el <code>AWS::Serverless::WebFunction</code> recurso	Se utiliza <code>AWS::Serverless::WebFunction</code> para definir una función web Lambda, que atiende las solicitudes HTTP a través de un punto final gestionado. Para obtener más información, consulte AWS::Serverless::WebFunction .	12 de octubre de 2026
Se agregaron OAuth 2.0 tipos de autenticación de IAM para las fuentes de SelfManagedKafka eventos	La <code>SourceAccessConfigurations</code> propiedad de una fuente de SelfManagedKafka eventos ahora acepta los mecanismos de <code>IAM_OAUTHBEARER_AUTH</code> autenticación <code>OAUTHBEARER_AUTH</code> <code>IAM_AUTH</code> , y los tipos de <code>OAUTHBEARER_IDENTITY_POOL</code> metadatos <code>OAUTHBEARER_SCOPE</code> <code>OAUTHBEARER_AUDIENCE</code> <code>OAUTHBEARER_LOGICAL_CLUSTER</code> , y. Para obtener más informaci	8 de octubre de 2026

	ón, consulte SelfManagedKafka .	
Se agregó la ManagedResourceTags propiedad para AWS::Serverless::CapacityProvider	Aplice etiquetas explícitas a las instancias de Amazon Elastic Compute Cloud y a otros recursos gestionados que un proveedor de capacidad aprovisiona en su nombre. Para obtener más información, consulte ManagedResourceTags .	25 de agosto de 2026
Se agregó la LoggingConfiguration propiedad para AWS::Serverless::CapacityProvider	Configure el registro del sistema para un proveedor de capacidad, de modo que la actividad de escalado y los eventos operativos vayan a un grupo de CloudWatch registros de Amazon en un nivel de registro configurable. Para obtener más información, consulte AWS::Serverless::CapacityProvider .	24 de agosto de 2026
Se agregó la opción --output para los comandos AWS SAMCLI	Especifique --output json que se imprima una salida estructurada y legible por máquina que pueda consumir mediante programación, por ejemplo, en CI/CD canalizaciones y extensiones IDE. La opción está disponible para sam build, samdeploy y sam init.	19 de agosto de 2026

[Se agregó la StorageMode propiedad para el código de la función y el contenido de la capa](#)

Controle si Lambda almacena una copia de su paquete de implementación o si hace referencia a ella desde su bucket de Amazon S3. Para obtener más información, consulte [FunctionCode](#) y [LayerContent](#).

4 de agosto de 2026

[Se actualizó AWS SAMCLI el soporte para crear Rust funciones para adaptarlo a la disponibilidad general](#)

La creación de Rust funciones ya Cargo Lambda no requiere la suscripción a las funciones beta. La guía para proyectos que contienen múltiples funciones ahora utiliza un Cargo espacio de trabajo. Para obtener más información, consulte [Creación de funciones Rust Lambda](#) con. Cargo Lambda

3 de agosto de 2026

[Se agregó AWS SAMCLI soporte para las versiones Java 8, Java 11 y Java 17 en AL2023](#)

java8.al2023 java11.al2023 , y ahora java17.al2023 están disponibles como tiempos de ejecución y repositorios de imágenes. Consulte [Repositorios de imágenes](#) y [sam](#) init.

13 de julio de 2026

[Se agregaron los tipos AWS::Serverless::NetworkConnector de recursos AWS::Serverless::MicrovmImage y](#)

Las nuevas secciones de referencia describen cómo crear una imagen de microVM de Lambda a partir de un artefacto de código almacenado en Amazon S3 y cómo definir la conectividad de red para los recursos informáticos de Lambda. Para obtener más información, consulte [AWS::Serverless::MicrovmImage](#) y [AWS::Serverless::NetworkConnector](#).

6 de julio de 2026

[Se agregó la opción --express para los AWS SAMCLI comandos sam deploy y sam sync](#)

Utilice el modo CloudFormation Express para que las operaciones de apilamiento se completen cuando se aplique la configuración de los recursos, mientras los recursos siguen preparándose en segundo plano. Para obtener más información, consulte [sam deploy](#) y [sam sync](#).

1 de julio de 2026

[Se agregó la opción --no-watch para los comandos AWS SAMCLI sam local start-api y sam local start-lambda](#)

Desactiva la visualización de archivos cuando la utilices --warm-containers, en el caso de proyectos en los que la vigilancia suponga una sobrecarga o sobrecarga de la CPU. I/O Para obtener más información, consulta [sam local start-api y sam local start-lambda](#).

10 de junio de 2026

[Se agregó AWS SAMCLI soporte para la AWS::LanguageExtensions transformación](#)

AWS SAMCLIAhora se expandeFn::ForEach , Fn::Length Fn::ToJsonString , y Fn::FindInMap de DefaultValue forma localsam build, por lo que sam packagesam deploy,sam sync,sam validate, y los sam local comandos funcionan con plantillas que utilizan la AWS::LanguageExtensions transformación. Para obtener más información, consulta la [compatibilidad con extensiones de CloudFormation idioma](#).

13 de mayo de 2026

[Se agregó AWS SAMCLI soporte para Ruby 4.0](#)

RubyLa versión 4.0 ya está disponible como un repositorio de imágenes y de tiempo de ejecución. Consulte [Repositorios de imágenes](#) y [sam](#) init.

27 de abril de 2026

[Se agregó el tipo AWS::Serverless::WebSocketApi de recurso](#)

Defina una WebSocket API configuración que incluya la ruta, la autorización, el dominio personalizado y la configuración de Route 53. AWS SAM Para obtener más información, consulte [AWS::Serverless::WebSocketApi](#).

13 de abril de 2026

Se ha ampliado la opción <code>--invoke-image</code> para los comandos locales AWS SAMCLI	Especifique <code>--invoke-image</code> más de una vez para usar una imagen de contenedor diferente por recurso y <code>ImageUri</code> anule las funciones que la usen. <code>PackageType: Image</code> Para obtener más información, consulta sam local invoke .	10 de abril de 2026
Se agregaron las <code>EndpointAccessMode</code> propiedades <code>SecurityPolicy</code> y para <code>AWS::Serverless::Api</code>	Configure la política de TLS seguridad y el modo de acceso al punto final en un dominio personalizado. Para obtener más información, consulte DomainConfiguration .	10 de abril de 2026
Se agregó la propiedad de <code>DockerImage</code> compilación para las compilaciones de imágenes de contenedores	Especifique un nombre personalizado para la Docker imagen AWS SAMCLI creada. Para obtener más información, consulte Cómo crear una aplicación .	10 de abril de 2026
Se agregó la opción <code>--container-dns</code> para los comandos locales AWS SAMCLI	Especifique los DNS servidores que utilizan los contenedores locales. La opción está disponible para sam local invoke , sam local start-api y sam local start-lambda .	10 de abril de 2026

Se ha añadido compatibilidad con el sistema de archivos Amazon S3 Files para AWS::Serverless::Function	Adjunte un punto de acceso Amazon S3 Files a una función con <code>FileSystemConfigs</code> , además de un punto de acceso Amazon EFS. Para obtener más información, consulte AWS::Serverless::Function .	8 de abril de 2026
Se agregó compatibilidad con la CLI de AWS SAM para Docker BuildKit	Cree funciones de imagen de contenedores con <code>BuildKit</code> . Para obtener más información, consulte Uso de sam build .	10 de marzo de 2026
Se agregó soporte de .env archivos para la opción --env-vars	La <code>--env-vars</code> opción de los <code>sam local</code> comandos ahora acepta un <code>.env</code> archivo además de JSON. Para obtener más información, consulte Invocar funciones localmente .	6 de marzo de 2026
Se agregó la <code>IpAddressType</code> propiedad para <code>AWS::Serverless::Api</code>	Configure un punto final de API como <code>ipv4 dualstack</code> . Para obtener más información, consulte EndpointConfiguration .	20 de febrero de 2026
Se agregó AWS SAM CLI soporte para el administrador de paquetes	Cree Python funciones con <code>uv</code> . Para obtener más información, consulte Crear Python funciones con uv .	19 de febrero de 2026

[Se agregó la `ResponseTransferMode` propiedad para eventos](#)

Transmita la respuesta de una función `ResponseTransferMode` configurando en `RESPONSE_STREAM`. Para obtener más información, consulte [Api](#).

11 de febrero de 2026

[Se agregaron las `MetricsConfig` propiedades `LoggingConfig` y para los eventos de Amazon MSK](#)

Configure el registro de mapeo de fuentes de eventos y las métricas mejoradas para una fuente de eventos de Amazon MSK. Para obtener más información, consulte [MSK](#).

2 de febrero de 2026

[Se agregó la `TenancyConfig` propiedad para `AWS::Serverless::Function`](#)

Configure el aislamiento de inquilinos para una función y transfiera un inquilino con la nueva `--tenant-id` opción activada `local invoke` y `remote invoke`. Para obtener más información, consulte [AWS::Serverless::Function](#).

20 de noviembre de 2025

[Se agregó la `ProvisionedPollerConfig` propiedad para los eventos de Amazon SQS](#)

Utilice el modo aprovisionado para una fuente de eventos de Amazon SQS estableciendo un número mínimo y máximo de sondeadores. Para obtener más información, consulte [SQS](#).

20 de noviembre de 2025

Se agregaron propiedades de reintento y lote para los eventos SelfManagedKafka

La fuente de eventos de Kinesis autogestionada ahora admite `BisectBatchOnFunctionError`, `FunctionResponseTypes`, `MaximumRecordAgeInSeconds`, y `MaximumRetryAttempts`. Para obtener más información, consulte [SelfManagedKafka](#).

20 de noviembre de 2025

Se agregó AWS SAM CLI soporte para Node.js 24

Node.js 24 ya está disponible como un repositorio de imágenes y de tiempo de ejecución. Consulte [Repositorios de imágenes](#) y `sam init`.

19 de noviembre de 2025

Se agregaron propiedades de reintento y lote para los eventos de Amazon MSK

La fuente de eventos de Amazon MSK ahora admite `BisectBatchOnFunctionError`, `FunctionResponseTypes`, `MaximumRecordAgeInSeconds`, y `MaximumRetryAttempts`. Para obtener más información, consulte [MSK](#).

19 de noviembre de 2025

[Se agregó el tipo AWS::Serverless::CapacityProvider de recurso](#)

Ejecute funciones en la capacidad de Amazon Elastic Compute Cloud de su propiedad y asocie un proveedor de capacidad a una función con `CapacityProviderConfig`. Para obtener más información, consulte [AWS::Serverless::CapacityProvider](#).

17 de noviembre de 2025

[Se agregó AWS SAM CLI soporte para Java 25 y Python 3.14](#)

JavaLas versiones 25 y Python 3.14 ya están disponibles como repositorios de imágenes e `sam init` imágenes base. Consulte [Repositorios de imágenes](#) y `sam init`.

10 de noviembre de 2025

[Se agregó la `Enabled` propiedad para los eventos de Amazon MSK](#)

Detenga y reanude un mapeo de origen de eventos de Amazon MSK. Para obtener más información, consulte [MSK](#).

16 de octubre de 2025

[Se agregaron propiedades de compilación que preservan la estructura Python del paquete](#)

Especifique `ParentPackageMode` y `ParentPackages` debajo `BuildProperties` para que la función Lambda creada pueda utilizar importaciones absolutas. Para obtener más información, consulte [Cómo crear una aplicación](#).

6 de octubre de 2025

[Se agregó AWS SAM soporte para funciones duraderas](#)

Configure una función duradera con la `DurableConfig` propiedad y pruebe una localmente con el nuevo `sam local callback` comando. Para obtener más información, consulte [DurableConfigy Prueba de funciones duraderas.](#)

25 de septiembre de 2025

[Se agregó compatibilidad con la CLI de AWS SAM para Finch](#)

Utilícela en lugar de Docker `compilar --use-container` y probar su aplicación localmente. Para obtener más información, consulte [Instalación Finch.](#)

19 de agosto de 2025

[Se agregó la BatchSize propiedad para los eventos de Amazon MSK](#)

Establezca el número máximo de registros que Lambda recupera en un solo lote desde una fuente de eventos de Amazon MSK. Para obtener más información, consulte [MSK.](#)

5 de agosto de 2025

[Se agregó una guía para convertir una función de consola Lambda en AWS SAM](#)

Mueva una función que creó en la consola de Lambda a una AWS SAM plantilla para poder administrarla tanto como infraestructura como código. Para obtener más información, consulte [Convertir funciones Lambda en aplicaciones. AWS SAM](#)

5 de agosto de 2025

Se agregó la SchemaRegistryConfig propiedad para los eventos de Kinesis	Utilice un registro de esquemas con una fuente de eventos de Amazon MSK o Kinesis autogestionada. Para obtener más información, consulte MSK y SelfManagedKafka .	19 de junio de 2025
Se agregó AWS SAM CLI soporte para .NET 9	.NET 9 ya está disponible como repositorio de imágenes. Consulte Repositorios de imágenes .	4 de junio de 2025
Se agregó AWS SAM compatibilidad con nombres de dominio personalizados en las API privadas	Adjunta un dominio personalizado a una API REST privada con las DomainAccessAssociation propiedades nuevas Policy y. Para obtener más información, consulte DomainConfiguration y DomainAccessAssociation .	4 de abril de 2025
Se agregó AWS SAM CLI soporte para Ruby 3.4	RubyLa versión 3.4 ya está disponible como un repositorio de imágenes y de tiempo de ejecución. Consulte Repositorios de imágenes y sam init.	24 de marzo de 2025
Se agregó la opción --runtime para el comando AWS SAM CLI sam local invoke	Anule el tiempo de ejecución definido en su plantilla para probar una función en un tiempo de ejecución diferente. Para obtener más información, consulta sam local invoke .	11 de marzo de 2025

[Se agregaron las opciones --mount-symlinks y --no-memory-limit para los comandos AWS SAMCLI](#)

Monta los enlaces simbólicos de nivel superior en el contenedor de compilación con [sam build](#) y [sam local invoke](#), y elimina --mount-symlinks el límite de memoria del contenedor con los comandos. --no-memory-limit [sam local](#)

26 de febrero de 2025

[Se agregó la opción --no-use-container para los comandos AWS SAMCLI sam build y sam sync](#)

Cree su función en la máquina local en lugar de dentro de un contenedor. Para obtener más información, consulta [sam build](#) y [sam sync](#).

9 de diciembre de 2024

[Se agregó AWS SAMCLI soporte para Node.js 2.2 y Python 3.13](#)

Node.js Las versiones 22 y Python 3.13 ya están disponibles como entornos de ejecución y repositorios de imágenes. Consulte [Repositorios de imágenes](#) y [sam init](#).

5 de diciembre de 2024

[Se agregó la PublishLambdaVersion propiedad para AWS::Serverless::LayerVersion](#)

Controle si AWS SAM publica una nueva versión de capa. Para obtener más información, consulte [AWS::Serverless::LayerVersion](#).

5 de diciembre de 2024

[Se agregaron las `MetricsConfig` propiedades `ProvisionedPollerConfig` y para las asignaciones de fuentes de eventos](#)

Utilice el modo aprovisionado para Amazon MSK y las fuentes de eventos de Kinesis autogestionadas y opte por las métricas de mapeo de fuentes de eventos mejoradas para las fuentes de eventos de DynamoDB, Kinesis y Amazon SQS `ProvisionedPollerConfig` con `MetricsConfig`. Para obtener más información, consulte [MSK](#) y [SQS](#).

25 de noviembre de 2024

[Se agregó la propiedad para `SourceKMSKeyArn` `AWS::Serverless::Function`](#)

Especifique la AWS KMS clave que se AWS SAM utiliza para cifrar el paquete de `.zip` despliegue. Para obtener más información, consulte [AWS::Serverless::Function](#).

14 de noviembre de 2024

[Se agregó la `RecursiveLoop` propiedad para `AWS::Serverless::Function`](#)

Defina si Lambda termina o permite un bucle de invocación de funciones recursivas. Para obtener más información, consulte [AWS::Serverless::Function](#).

27 de agosto de 2024

[Se agregó la `KmsKeyArn` propiedad para las asignaciones de fuentes de eventos](#)

Especifique una AWS KMS clave que cifre los criterios de filtro de mapeo de fuentes de eventos inactivos. La propiedad está disponible en las fuentes de eventos de Amazon DocumentDB, DynamoDB, Kinesis, Amazon MQ, Amazon MSK y Amazon SQS. SelfManagedKafka Para obtener más información, consulte [MSK](#).

21 de agosto de 2024

[UseAliasAsEventTarget Se agregó la propiedad para `AWS::Serverless::StateMachine`](#)

Utilice el alias creado por `AutoPublishAlias` como destino del origen del evento. Para obtener más información, consulte [AWS::Serverless::StateMachine](#).

12 de agosto de 2024

[Se reestructuró y actualizó contenido en toda la guía para desarrolladores](#)

Se reorganizó y reestructuró la guía para mejorar la visibilidad y la usabilidad. Se actualizaron y mejoraron los títulos. Se aportaron detalles adicionales al presentar los temas y conceptos.

20 de junio de 2024

[Se agregó compatibilidad de la CLI de AWS SAM con Ruby 3.3](#)

Ruby 3.3 ya está disponible como repositorio de imágenes y de versión ejecutable. Consulte [Repositorios de imágenes](#) y [sam init](#) para obtener más información.

4 de abril de 2024

[Se agregaron opciones de comando de la CLI de AWS SAM](#)

Hay nuevas opciones disponibles para el comando [sam local start-api](#): `--ssl-cert-file PATH` , `--ssl-key-file PATH` . Además, la nueva opción de línea de comandos `--add-host LIST` está disponible para [sam local invoke](#), [sam local start-api](#) y [sam local start-lambda](#)

20 de marzo de 2024

[Se agregó compatibilidad con la CLI de AWS SAM para .NET 8](#)

.NET 8 ya está disponible como repositorio de imágenes y de versión ejecutable. Ya no se admiten los tiempos de ejecución y los repositorios de imágenes para .NET Core 3.1, Node.js 14, Node.js 12, Python 3.7 y Ruby 2.7. Consulte [Repositorios de imágenes](#) y [sam init](#).

22 de febrero de 2024

[Se agregó el instalador del paquete de arm64 de la AWS SAM de CLI para Linux](#)

Para obtener instrucciones, consulte [Instalación del AWS SAM CLI](#).

6 de diciembre de 2023

[Se agregó la opción `--watch-exclude` para el comando `sam sync` de la CLI de AWS SAM](#)

Excluye archivos y carpetas de modo que no inicien una sincronización. Para obtener más información, consulta [Especificar archivos y carpetas que no iniciarán una sincronización](#).

6 de diciembre de 2023

[Se agregó la opción --build-in-source para el comando sam sync de la CLI de AWS SAM](#)

Compila el proyecto en la carpeta de origen para acelerar el proceso de compilación. Para obtener más información, consulta [Acelera los tiempos de compilación al compilar el proyecto en la carpeta de origen.](#)

6 de diciembre de 2023

[Se agregó la opción --build-in-source para el comando sam build de la CLI de AWS SAM](#)

Compila el proyecto en la carpeta de origen para acelerar el proceso de compilación. Para obtener más información, consulta [Acelera los tiempos de compilación al compilar el proyecto en la carpeta de origen.](#)

6 de diciembre de 2023

[Se agregó compatibilidad con nuevos recursos para el comando remote invoke de la CLI de AWS SAM](#)

Usa `sam remote invoke` con las aplicaciones de Kinesis Data Streams, las colas de Amazon SQS y las máquinas de estado de Step Functions. Para obtener más información, consulta [Uso de la invocación remota de sam.](#)

15 de noviembre de 2023

[Se agregó el nuevo comando `remote test-event` de la CLI de AWS SAM para los eventos de prueba compatibles](#)

Úselo AWS SAM CLI para acceder y gestionar los eventos de prueba que se pueden compartir desde el registro de EventBridge esquemas para probar las funciones de Lambda en el. Nube de AWS Para obtener más información, consulta [Uso del comando `sam remote test-event`](#).

3 de octubre de 2023

[La compatibilidad de la CLI de AWS SAM con Terraform ya está disponible con carácter general](#)

Para obtener más información sobre la compatibilidad de la CLI de AWS SAM con Terraform, consulta [compatibilidad de la CLI de AWS SAM para Terraform](#).

5 de septiembre de 2023

[Se agregó compatibilidad con la CLI de AWS SAM para Terraform Cloud](#)

La AWS SAM CLI ahora es compatible con los tests locales para Terraform Cloud. Para obtener más información, consulta [Configurar para Terraform Cloud](#).

5 de septiembre de 2023

[Se agregó compatibilidad con el formato de archivo YAML para el archivo de configuración de la CLI de AWS SAM](#)

AWS SAMCLIAhora es compatible con el formato de archivo `[.yaml|.yml]`. [Se han actualizado las páginas de configuración de la CLI de AWS SAM](#) y del archivo de configuración de la [CLI de AWS SAM](#).

18 de julio de 2023

Se agregó soporte de AWS SAM CLI sam local start-api comandos para Terraform	El ¿Para qué AWS SAM CLI sirve el soporte Terraform? la sección se ha actualizado para incluir AWS SAM CLI sam local start-api el soporte de comandos para Terraform.	6 de julio de 2023
Se agregó un nuevo comando de invocación remota de la CLI de AWS SAM	Para empezar a usar sam remote invoke, consulta Uso de invocación remota sam .	22 de junio de 2023
Se agregó el tipo de recurso sin servidor AWS AppSync GraphQL API	Cree una nueva AWS::Serverless::GraphQLApi _sección que describa cómo definir un GraphQL API recurso con AWS SAM.	22 de junio de 2023
Se agregó compatibilidad con la CLI de AWS SAM para Ruby 3.2	Actualiza la página sam init para incluir una nueva imagen base y valores de tiempo de ejecución. Actualiza la página Repositorios de imágenes con Amazon ECR URI Ruby 3.2.	6 de junio de 2023
Se agregaron pasos opcionales para la verificación de la integridad del instalador del paquete de la CLI de AWS SAM	Actualiza la página Instalación de la AWS SAM CLI para que refleje el paso opcional. Crea la página Verificar la integridad del instalador de la CLI de AWS SAM para documentar los pasos.	31 de mayo de 2023

Se agregó la opción <code>sam sync</code> para omitir la sincronización de la infraestructura	Personalice si se requiere una CloudFormation implementación cada vez <code>sam sync</code> que se ejecute. Para obtener más información, consulte Omitir la CloudFormation implementación inicial .	23 de marzo de 2023
Se agregó compatibilidad con el tipo de origen de eventos de DocumentDB	La especificación AWS SAM de la plantilla ahora admite DocumentDB el tipo de fuente de eventos para el <code>AWS::Serverless::Function</code> recurso. Para obtener más información, consulta DocumentDB .	10 de marzo de 2023
Creación de funciones de Lambda Rust con Cargo Lambda	Usa la AWS SAM CLI para crear sus funciones de Lambda Rust utilizando Cargo Lambda. Para obtener más información, consulta Creación de funciones de Lambda Rust con Cargo Lambda .	23 de febrero de 2023
Cree recursos de funciones fuera de AWS SAM	Se ha añadido una guía sobre cómo omitir funciones al usar el comando <code>sam build</code> . Para obtener más información, consulte Crear funciones fuera de AWS SAM .	14 de febrero de 2023

[Nueva sintaxis de conectores integrados](#)

Usa la nueva sintaxis de conectores integrados para definir tus recursos `AWS::Serverless::Connector` . Para obtener más información, consulte [Administrar los permisos de recursos con AWS SAM conectores](#).

8 de febrero de 2023

[Se ha añadido un nuevo comando sam list para la CLI de AWS SAM](#)

Usa `sam list` para ver información importante sobre los recursos de la aplicación sin servidor. Para obtener más información, consulta [sam list](#).

2 de febrero de 2023

[Se agregaron propiedades de formato y OutExtension compilación para esbuild](#)

Node.js La creación de funciones Lambda con esbuild ahora admite `Format` y `OutExtension` crea propiedades. Para obtener más información, consulte [Creación de funciones Node.js Lambda con esbuild](#).

2 de febrero de 2023

[Se agregaron opciones de administración del tiempo de ejecución a la especificación de la plantilla AWS SAM](#)

Configura las opciones de administración del tiempo de ejecución para las funciones de Lambda. Para obtener más información, consulta [RuntimeManagementConfig](#) .

24 de enero de 2023

Se agregó la propiedad de destino al StateMachine recurso EventSource for AWS::Serverless::.	El tipo de recurso <code>AWS::Serverless::StateMachine</code> admite la propiedad <code>Target</code> para EventBridgeRule y los orígenes de eventos Schedule .	13 de enero de 2023
Configurar el escalado de los sondeadores de SQS para funciones de Lambda	Configura el escalado de los sondeadores de SQS con la propiedad <code>ScalingConfig</code> para <code>AWS::Serverless::Function</code> . Para obtener más información, consulta ScalingConfig .	12 de enero de 2023
Valide AWS SAM las aplicaciones con cfn-lint	Puede usar <code>cfn-lint</code> para validar sus plantillas a través del <code>AWS SAM CLI</code> . Para obtener más información, consulta Validar con cfn-lint .	11 de enero de 2023
Supervise sus aplicaciones sin servidor con Application Insights CloudWatch	Configure Amazon CloudWatch Application Insights para supervisar sus AWS SAM aplicaciones. Para obtener más información, consulte Supervise sus aplicaciones sin servidor con CloudWatch Application Insights .	19 de diciembre de 2022
Se ha añadido un instalador de paquetes de la CLI de AWS SAM para macOS	Instala la AWS SAM CLI con el nuevo instalador del paquete macOS. Para obtener más información, consulte Instalación del AWS SAM CLI .	6 de diciembre de 2022

[Se agregó soporte para Lambda SnapStart](#)

Configure SnapStart las funciones de Lambda para crear instantáneas, que son estados en caché de las funciones inicializadas. Para obtener más información, consulte [AWS::Serverless::Function](#).

28 de noviembre de 2022

[Se ha añadido compatibilidad de la CLI de AWS SAM para nodejs18.x](#)

AWS SAM CLI ahora es compatible con el tiempo de ejecución nodejs18.x. Para obtener más información, consulta [sam init](#).

17 de noviembre de 2022

[Se ha añadido una guía sobre la configuración del acceso y los permisos](#)

AWS SAM [ofrece dos opciones que simplifican la administración del acceso y los permisos de las aplicaciones sin servidor](#). Para obtener más información, consulte [Administrar el acceso a los recursos y los permisos](#).

17 de noviembre de 2022

[Se ha añadido soporte para crear funciones de Lambda de .NET 7 con compilación AOT nativa](#)

Cree y empaquete sus funciones Lambda de .NET 7 utilizando la compilación nativa Ahead-of-Time (AOT) para mejorar los tiempos de arranque en frío de Lambda. AWS SAM Para obtener más información, consulta [Creación de funciones de Lambda de .NET 7 con la compilación AOT nativa](#).

15 de noviembre de 2022

[Se agregó AWS SAMCLITerraform soporte para la depuración y las pruebas locales](#)

Usa la AWS SAM CLI en sus proyectos de Terraform para realizar depuraciones y testeos locales de sus funciones y capas de Lambda. Para obtener más información, consulta [Soporte de Terraform para la CLI de AWS SAM](#).

14 de noviembre de 2022

[Se agregó AWS SAM soporte para Scheduler EventBridge](#)

La especificación de la plantilla AWS Serverless Application Model (AWS SAM) proporciona una sintaxis simple y abreviada que puedes usar para programar eventos con EventBridge Scheduler para y. AWS Lambda AWS Step Functions Para obtener más información, consulte [Programar eventos con EventBridge Scheduler](#).

10 de noviembre de 2022

[Se simplificaron las instrucciones de instalación de la CLI de AWS SAM](#)

Los requisitos previos de la CLI de AWS SAM y los pasos opcionales se han trasladado a páginas separadas. Los pasos de instalación de los sistemas operativos compatibles se encuentran en [Instalar el AWS SAMCLI](#).

4 de noviembre de 2022

[Se ha añadido una corrección para permitir rutas de acceso largas para los usuarios de Windows 10](#)

El repositorio de plantillas de aplicaciones de la CLI de AWS SAM contiene algunas rutas de archivo largas que puedes provocar errores al ejecutarse `sam init` debido a las limitaciones `MAX_PATH` de Windows 10. Para obtener más información, consulta [Instalación de la CLI de AWS SAM](#).

4 de noviembre de 2022

[Se actualizó el proceso de implementación gradual para las primeras implementaciones](#)

La implementación gradual de una función Lambda AWS CodeDeploy requiere dos pasos. Para obtener más información, consulta [Implementación gradual de una función de Lambda por primera vez](#).

13 de octubre de 2022

[Soporte adicional de filtrado de eventos de Lambda para más tipos de eventos](#)

`FilterCriteria` propiedad es añadidas a tipos de orígenes de eventos [MSK](#), [MQ](#) y [SelfManagedKafka](#).

13 de octubre de 2022

[Se agregó compatibilidad con OpenID Connect \(OIDC\) para la canalización AWS SAM](#)

AWS SAM admite la autenticación de usuarios de OpenID Connect (OIDC) para Bitbucket, GitHub Actions y plataformas de integración GitLab continua y entrega continua (.). CI/CD Para obtener más información, consulta [Cómo usar cuentas de usuario de OIDC](#) con Pipeline. AWS SAM

13 de octubre de 2022

[Nota sobre las propiedades JwtConfiguration](#)

Se ha añadido una nota sobre la definición de las propiedades issuer y audience bajo JwtConfiguration para [OAuth2Authorizer](#).

7 de octubre de 2022

[Nuevas propiedades para Function y StateMachine EventSource](#)

Propiedades Enabled y State agregadas al origen de eventos CloudWatchEvent para [AWS::Serverless::Function](#). Propiedad State agregada al origen de eventos Schedule para [AWS::Serverless::Function](#) y [AWS::Serverless::StateMachine](#).

6 de octubre de 2022

[AWS SAM los conectores ahora están disponibles de forma general](#)

Los conectores son un tipo de recurso AWS SAM abstracto, identificado como `AWS::Serverless::Connector`, que proporciona un método sencillo y seguro de aprovisionar permisos entre los recursos de aplicaciones sin servidor. Para obtener más información, consulte [Administrar los permisos de recursos con AWS Serverless Application Model conectores](#).

6 de octubre de 2022

[Se han añadido nuevas opciones de sincronización de sam a la CLI de AWS SAM](#)

Se han añadido opciones `--dependency-layer` y `--use-container` a [sam sync](#).

20 de septiembre de 2022

Se han añadido nuevas opciones de sincronización de sam a la CLI de AWS SAM	Opción <code>--on-failure</code> agregada a sam deploy .	9 de septiembre de 2022
La compatibilidad esbuild ya está disponible con carácter general	Para crear y empaquetar funciones Node.js Lambda, puede utilizarlas AWS SAMCLI con el paquete JavaScript esbuild .	1 de septiembre de 2022
Telemetría actualizada de la CLI de AWS SAM	La descripción de la información del sistema y el entorno recopilada se ha actualizado para incluir los valores hash de los atributos de uso.	1 de septiembre de 2022
Se ha añadido compatibilidad con variables de entorno local para la CLI de AWS SAM	Usa variables de entorno de la AWS SAM CLI cuando invoques funciones de Lambda de forma local y cuando ejecuta API Gateway de forma local .	1 de septiembre de 2022

[Compatibilidad con arquitecturas del conjunto de instrucciones de Lambda](#)

Usa la AWS SAM CLI para crear funciones de Lambda y capas de Lambda para arquitecturas de conjuntos de instrucciones x86_64 o arm64. Para obtener más información, consulte la propiedad [Architectures](#) del tipo de `AWS::Serverless::Function` recurso y la [CompatibleArchitectures](#) propiedad del tipo de recurso. `AWS::Serverless::LayerVersion`

1 de octubre de 2021

[Generación de ejemplos de configuraciones de canalización](#)

Utilice la AWS SAM CLI para generar canalizaciones de ejemplo para varios CI/CD sistemas mediante los comandos `new sam pipeline bootstrap` y `new sam pipeline init`. Para obtener más información, consulte [Generación de CI/CD canalizaciones de ejemplo](#).

21 de julio de 2021

[Integración de AWS CDK de la CLI de AWS SAM\(versión preliminar, fase 2\)](#)

Con la fase 2 de la versión preliminar pública, ahora puede utilizarla AWS SAMCLI para empaquetar e implementar AWS CDK aplicaciones. También puede descargar una AWS CDK aplicación de muestra directamente mediante AWS SAMCLI. Para obtener más información, consulta [Kit de desarrollo de la nube de AWS \(AWS CDK\) \(versión preliminar\)](#).

13 de julio de 2021

[Compatibilidad con RabbitMQ como origen de eventos para funciones](#)

Se ha añadido compatibilidad con RabbitMQ como origen de eventos para funciones sin servidor. Para obtener más información, consulta la propiedad [SourceAccessConfigurations](#) del origen de eventos MQ del tipo de recurso [AWS::Serverless::Function](#).

7 de julio de 2021

[Implementación de aplicaciones sin servidor mediante imágenes de contenedores de compilación de Amazon ECR](#)

Utilice imágenes de contenedores de compilación de Amazon ECR para implementar aplicaciones sin servidor con CI/CD sistemas comunes como AWS CodePipeline Jenkins y Actions GitLab CI/CD. GitHub Para obtener más información, consulta [Implementación de aplicaciones sin servidor](#).

24 de junio de 2021

Depuración AWS SAM de aplicaciones con kits de herramientas AWS	AWS Los kits de herramientas ahora admiten la depuración gradual con más combinaciones de entornos de desarrollo o integrados (IDE) y tiempos de ejecución. Para obtener más información, consulte Uso de kits de herramientas. AWS	20 de mayo de 2021
Integración de AWS CDK de la CLI de AWS SAM(versión preliminar)	Ahora puede usarlo AWS SAMCLI para probar y crear AWS CDK aplicaciones de forma local. Esta es una versión preliminar pública. Para obtener más información, consulta Kit de desarrollo de la nube de AWS (AWS CDK) (versión preliminar) .	29 de abril de 2021
El repositorio de imágenes del contenedor predeterminado cambió a Amazon ECR Public	El repositorio de imágenes del contenedor predeterminado pasó de DockerHub ser Amazon ECR Public . Para obtener más información, consulta Repositorios de imágenes .	6 de abril de 2021
Compilaciones nocturnas de la CLI de AWS SAM	Ahora puede instalar una versión preliminar del AWS SAMCLI, que se crea todas las noches. Para obtener más información, consulte la sección Compilación nocturna del subtema del sistema operativo que prefiera en Instalación del. AWS SAMCLI	25 de marzo de 2021

[Compatibilidad con variables del entorno de compilación de contenedores](#)

Ahora puedes pasar variables de entorno para crear contenedores. Para obtener más información, consulta las opciones `--container-env-var` y `--container-env-var-file` en [sam build](#).

4 de marzo de 2021

[Nuevo proceso de instalación de Linux](#)

Ahora puedes instalar la AWS SAM CLI con un instalador nativo de Linux. Para obtener más información, consulta [Instalación de la CLI de AWS SAM en Linux](#).

10 de febrero de 2021

[Support for dead-letter colas para EventBridge](#)

Se ha añadido soporte para colas de texto sin formato y fuentes de Schedule eventos para funciones sin EventBridge servidor y máquinas de estado. Para obtener más información, consulta la propiedad `DeadLetterConfig` del origen de eventos `EventBridgeRule` y Schedule para los tipos de recursos [AWS::Serverless::Function](#) y [AWS::Serverless::StateMachine](#).

29 de enero de 2021

Compatibilidad con los puntos de control personalizados

Se ha añadido compatibilidad con puntos de control personalizados para los orígenes de eventos de DynamoDB y Kinesis para funciones sin servidor. Para obtener más información, consulta la propiedad `FunctionResponseType` de los tipos de datos [Kinesis](#) y [DynamoDB](#) del tipo de recurso [AWS::Serverless::Function](#).

29 de enero de 2021

Compatibilidad con ventanas de saltos de tamaño constante

Se ha añadido compatibilidad con ventanas de saltos para los orígenes de eventos de DynamoDB y Kinesis para funciones sin servidor. Para obtener más información, consulta la propiedad `TumblingWindowInSeconds` de los tipos de datos [Kinesis](#) y [DynamoDB](#) del tipo de recurso [AWS::Serverless::Function](#).

17 de diciembre de 2020

[Compatibilidad con los contenedores de calentamiento](#)

Se ha añadido compatibilidad con contenedores de calentamiento cuando se realizan testeos locales mediante los comandos [sam local start-api](#) y [sam local start-lambda](#) de la CLI de AWS SAM. Para obtener más información, consulta la opción `--warm-containers` de esos comandos.

16 de diciembre de 2020

[Compatibilidad con imágenes del contenedor de Lambda](#)

Se ha añadido compatibilidad con imágenes del contenedor de Lambda. Para obtener más información, consulta [Creación de aplicaciones](#).

1 de diciembre de 2020

[Compatibilidad con la firma de código](#)

Se ha añadido compatibilidad con la firma de código y la implementación confiable de código de aplicaciones sin servidor. Para obtener más información, consulte [Configuración de la firma de código](#) para aplicaciones AWS SAM.

23 de noviembre de 2020

Compatibilidad con compilaciones en paralelo y en caché

Se mejoró el rendimiento de las compilaciones de aplicaciones sin servidor al agregar dos opciones al comando [sam build](#): `--parallel`, que crea funciones y capas en paralelo en lugar de secuencialmente, y `--cached`, que usa artefactos de compilación de compilaciones anteriores cuando no se han realizado cambios que requieran una reconstrucción.

10 de noviembre de 2020

[Compatibilidad con Amazon MQ y autenticación TLS mutua](#)

Se ha añadido compatibilidad con Amazon MQ como origen de eventos para funciones sin servidor. Para obtener más información, consulte los tipos de datos [EventSource](#) y [MQ](#) del tipo de recurso [AWS::Serverless::Function](#).

También ha añadido soporte para la autenticación mutua de Seguridad de la capa de transporte (TLS) para las API de API Gateway y las API HTTP. Para obtener más información, consulta el tipo de datos [DomainConfiguration](#) del tipo de recurso [AWS::Serverless::Api](#) o el tipo de datos [HttpApiDomainConfiguration](#) del tipo de recurso [AWS::Serverless::HttpApi](#).

5 de noviembre de 2020

[Compatibilidad con los autorizadores de Lambda para las API HTTP](#)

Se ha añadido compatibilidad con los autorizadores de Lambda para el tipo de recurso [AWS::Serverless::HttpApi](#). Para obtener más información, consulta [Ejemplo de función de Lambda \(AWS::Serverless::HttpApi\)](#).

27 de octubre de 2020

[Compatibilidad con múltiples archivos y entornos de configuración](#)

Se ha añadido compatibilidad con varios entornos y archivos de configuración para almacenar los valores de los parámetros predeterminados de los comandos de la CLI de AWS SAM. Para obtener más información, consulta [Archivo de configuración de la CLI de AWS SAM](#).

24 de septiembre de 2020

[Support for X-Ray Step Functions y referencias a la hora de controlar el acceso a las API](#)

Se agregó soporte para X-Ray como fuente de eventos para máquinas de estado sin servidor. Para obtener más información, consulta la propiedad [Tracing](#) del tipo de recurso [AWS::Serverless::StateMachine](#). También ha añadido soporte para referencias al controlar el acceso a las API. Para obtener más información, consulta el tipo de datos [ResourcePolicyStatement](#).

17 de septiembre de 2020

[Compatibilidad con Amazon MSK](#)

Se ha añadido compatibilidad con Amazon MSK como origen de eventos para funciones sin servidor. Esto permite que los registros de un tema de Amazon MSK activen tu función de Lambda. Para obtener más información, consulta los tipos de datos [EventSource](#) y [MSK](#) y el tipo de recurso [AWS::Serverless::Function](#).

13 de agosto de 2020

[Compatibilidad con Amazon EFS](#)

Se ha añadido soporte para montar sistemas de archivos Amazon EFS en directorios locales. Esto permite al código de tu función de Lambda acceder a los recursos compartidos y también te permite modificarlos. Para obtener más información, consulta la propiedad [FileSystemConfigs](#) del tipo de recurso [AWS::Serverless::Function](#).

16 de junio de 2020

[Orquestación de aplicaciones sin servidor](#)

Se ha añadido soporte para orquestar aplicaciones mediante la creación de máquinas de estado de Step Functions mediante AWS SAM. Para obtener más información, consulte [Organizar AWS recursos con AWS Step Functions](#) y el tipo de [AWS::Serverless::StateMachine](#) recurso.

27 de mayo de 2020

[Creación de tiempos de ejecución personalizados](#)

Se ha añadido la capacidad de crear tiempos de ejecución personalizados. Para obtener más información, consulta acerca de la creación de un proveedor de identidad personalizado, consulta [Creación de tiempos de ejecución personalizados](#).

21 de mayo de 2020

[Creación de capas](#)

Se ha añadido la capacidad de crear recursos `LayerVersion` individuales. Para obtener más información, consulta [Creación de capas](#).

19 de mayo de 2020

[Recursos generados CloudFormation](#)

Proporcionó detalles sobre los CloudFormation recursos que AWS SAM genera y cómo hacer referencia a ellos. Para obtener más información, consulte [CloudFormation Recursos generados](#).

8 de abril de 2020

[Configuración de AWS credenciales](#)

Se agregaron instrucciones para configurar AWS las credenciales en caso de que aún no las haya configurado para usarlas con otras AWS herramientas, como uno de los AWS SDK o el AWS CLI. Para obtener más información, consulte [Configuración de AWS credenciales](#).

17 de enero de 2020

[AWS SAM especificación y AWS SAMCLI actualizaciones](#)

Se migró la AWS SAM especificación desde GitHub. Para obtener más información, consulta la [especificación de AWS SAM](#). También se actualizó el flujo de trabajo de implementación con cambios en el comando [sam deploy](#).

25 de noviembre de 2019

[Nuevas opciones para controlar el acceso a las API de API Gateway y a las actualizaciones de las plantillas de políticas](#)

Se han añadido nuevas opciones para controlar el acceso a las API de API Gateway: permisos de IAM, claves de API y políticas de recursos. Para obtener más información, consulta [Control del acceso a las API de API Gateway](#). También se actualizaron dos plantillas de políticas: RekognitionFacesPolicy y ElasticsearchHttpPolicy. Para obtener más información, consulta [Plantillas de la política de AWS SAM](#)

29 de agosto de 2019

[Introducción a las actualizaciones](#)

Se actualizó el capítulo de introducción con instrucciones de instalación mejoradas para la CLI de AWS SAM y el tutorial Hello World. Para obtener más información, consulte [Primeros pasos con AWS SAM](#).

25 de julio de 2019

[Control del acceso a las API de API Gateway](#)

Se ha añadido compatibilidad para controlar el acceso a las API de API Gateway. Para obtener más información, consulta [Control del acceso a las API de API Gateway](#).

21 de marzo de 2019

[Se ha añadido el `sam publish` a la CLI de AWS SAM](#)

El nuevo comando [`sam publish`](#) de la CLI de AWS SAM simplifica el proceso de publicación de aplicaciones sin servidor en AWS Serverless Application Repository. Para obtener más información, consulte [Publicar aplicaciones sin servidor mediante AWS SAMCLI](#).

21 de diciembre de 2018

[Compatibilidad con capas y aplicaciones anidadas](#)

Se ha añadido compatibilidad con capas y aplicaciones anidadas. Para obtener más información, consulta [Uso de aplicaciones anidadas](#) y [Trabajar con capas](#).

29 de noviembre de 2018

[Se ha añadido el `sam build` a la CLI de AWS SAM](#)

El nuevo comando [`sam build`](#) de la CLI de AWS SAM simplifica el proceso de compilación de aplicaciones sin servidor con dependencias para que pueda probarlas e implementarlas localmente. Para obtener más información, consulta [Creación de aplicaciones](#).

19 de noviembre de 2018

[Se han añadido nuevas opciones de instalación para la CLI de AWS SAM](#)

Se agregaron las opciones de instalación de Linuxbrew (Linux), MSI (Windows) y Homebrew (macOS) para AWS SAMCLI [Para obtener más información, consulte Instalación del AWS SAMCLI](#)

7 de noviembre de 2018

[Nueva guía](#)

Esta es la primera versión de la Guía para desarrolladores de AWS Serverless Application Model .

17 de octubre de 2018

Las traducciones son generadas a través de traducción automática. En caso de conflicto entre la traducción y la version original de inglés, prevalecerá la version en inglés.