



Guía para desarrolladores

AWS Serverless Application Model



AWS Serverless Application Model: Guía para desarrolladores

Copyright © 2024 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
Características principales	1
Información relacionada	2
Funcionamiento	2
¿Cuál es la especificación AWS SAM de la plantilla?	3
¿Qué son el AWS SAM proyecto y la AWS SAM plantilla?	3
¿Qué es el? AWS SAMCLI	10
Más información	17
Sigüientes pasos	18
Conceptos sobre sistemas sin servidor	18
Conceptos sobre sistemas sin servidor	18
Introducción	20
Requisitos previos	20
Paso 1: Crear una AWS cuenta	21
Paso 2: Crear una cuenta de usuario de IAM	21
Paso 3: Crear un ID de clave de acceso y la clave de acceso secreta	22
Paso 4: Instale el AWS CLI	24
Paso 5: utilícelo AWS CLI para configurar AWS las credenciales	24
Sigüientes pasos	25
Instale la CLI AWS SAM	25
Instalación de la CLI de AWS SAM	26
Solución de errores de la instalación de	37
Sigüientes pasos	38
Opcional: compruebe el AWS SAMCLI instalador	39
Tutorial Hello World	52
Requisitos previos	53
Paso 1: inicializar la aplicación de muestra Hello World	53
Paso 2: crear su aplicación	57
Paso 3: Implemente la aplicación en el Nube de AWS	59
Paso 4: ejecutar la aplicación	63
Paso 5: Interactúa con tu función en el Nube de AWS	65
Paso 6: Modifique y sincronice su aplicación con Nube de AWS	66
Paso 7: (opcional) probar su aplicación localmente	69
Paso 8: Elimine la aplicación del Nube de AWS	71

Resolución de problemas	72
Más información	72
¿Cómo usar AWS SAM	73
El AWS SAMCLI	73
Cómo se documentan los comandos CLI de AWS SAM.	74
Configuración de la AWS SAM de CLI	75
Comandos principales	82
El AWS SAM proyecto	83
Anatomía de la plantilla	84
Recursos y propiedades	93
Recursos de generados	419
Atributos de recursos compatibles	437
APIExtensiones de pasarela	439
Funciones intrínsecas	441
Desarrolle su aplicación	442
Crea tu aplicación	442
Inicialización de aplicación sin servidor	443
Opciones para el inicio de sesión	449
Resolución de problemas	449
Ejemplos	449
Más información	450
Sigüientes pasos	450
Defina su infraestructura	451
Defina los recursos de la aplicación	451
Configura el acceso	453
Controla el API acceso	536
Aumente la eficiencia con capas	549
Reutilice el código	552
Gestione eventos basados en el tiempo	555
Orquestación de aplicaciones	558
Configura la firma de código	560
Validar archivos AWS SAM de plantilla	563
Cree su aplicación	564
Introducción a sam build	565
Compilación predeterminada con AWS SAM	579
Personaliza tu construcción	587

Pruebe su aplicación	613
Introducción a sam local	613
Uso del comando sam local	614
Introducción a sam local generate-event	614
Introducción a sam local invoke	621
Introducción a sam local start-api	627
Introducción a sam local start-lambda	633
Invoca funciones de forma local	636
Archivo de variables de entorno	637
Capas	638
Más información	638
API Gateway ejecutado localmente	638
Archivo de variables de entorno	640
Capas	642
Pruebe con sam remote test-event	642
Configure la CLI de AWS SAM para utilizar sam remote test-event.	643
Uso del comando sam remote test-event	643
Uso de eventos de prueba compartibles	646
Gestion de eventos de prueba compartibles	646
Realice pruebas con sam remote invoke	648
Uso del comando sam remote invoke	649
Uso de las opciones del comando sam remote invoke	653
Configure el archivo de configuración de su proyecto	659
Ejemplos	659
Vínculos relacionados	675
Automatice las pruebas de integración	675
Genere cargas útiles de muestra	677
Depure su aplicación	678
Funciones de depuración local	678
Uso de kits AWS de herramientas	679
Se ejecuta AWS SAM localmente en modo de depuración	681
Pase varios argumentos de tiempo de ejecución	682
Valide con cfn-lint	682
Ejemplos	683
Más información	683
Implemente su aplicación y sus recursos	684

Introducción a sam deploy	684
Requisitos previos	685
Implementación de aplicaciones mediante sam deploy	685
Prácticas recomendadas	696
Opciones para una implementación de sam	696
Resolución de problemas	696
Ejemplos	696
Más información	705
Opciones de implementación	705
Cómo utilizarlos para realizar AWS SAMCLI una implementación manual	705
Implemente con sistemas y canalizaciones de CI/CD	706
Implementación gradual	706
Solución de problemas de las implementaciones mediante la CLI de AWS SAM	706
Más información	638
Implemente con sistemas y canalizaciones de CI/CD	707
¿Qué es una canalización?	708
Genere una canalización inicial	709
Personaliza las canalizaciones iniciales	715
Automatice sus despliegues	717
Utilice la autenticación OIDC	721
Cargue archivos locales en el momento de la implementación	724
Introducción a sam sync	733
Detecte y sincronice automáticamente los cambios locales en el Nube de AWS	734
Personalice los cambios locales que se sincronizan con el Nube de AWS	735
Preparar su aplicación en la nube para probarla y validarla.	735
Opciones para el comando de sincronización sam	736
Resolución de problemas	738
Ejemplos	739
Más información	745
Supervise su aplicación	746
Perspectivas de aplicación	746
Configurar CloudWatch Application Insights con AWS SAM	746
Sigüentes pasos	750
Trabajo con registros	750
Obtención de registros por pila AWS CloudFormation	751
Obtención de registros por nombre de función de Lambda	751

Rastreo de registros	751
Visualización de registros para un intervalo de tiempo específico	751
Filtrado de registros	751
Resaltado de errores	752
JSONbonita impresión	752
AWS SAM referencia	753
AWS SAM la especificación y la plantilla AWS SAM	753
Referencia de los comandos AWS SAM CLI	753
Plantillas de política de AWS SAM	754
Temas	754
AWS SAMCLI comandos	754
sam build	755
sam delete	761
sam deploy	762
sam init	768
sam list	771
sam local generate-event	780
sam local invoke	781
sam local start-api	786
sam local start-lambda	791
sam logs	796
sam package	800
sam pipeline bootstrap	803
sam pipeline init	808
sam publish	809
sam remote invoke	811
sam remote test-event	816
sam sync	823
sam traces	830
sam validate	831
AWS SAMCLIGestión	833
Archivo de configuración de la CLI de AWS SAM	833
Administración de versiones de la CLI de AWS SAM	840
Configuración de las credenciales de AWS	849
Telemetría AWS SAM CLI	851
Resolución de problemas	853

Referencia de conector	859
Tipos de recursos de conector admitidos	859
Políticas de IAM creadas por conectores	870
Instalar Docker	893
Instalación de Docker	894
Sigüientes pasos	897
Repositorios de imágenes	897
Repositorio de imágenes URIs	898
Ejemplos	899
Implementación gradual	900
Implementación gradual de una función de Lambda por primera vez.	903
Más información	904
Notas importantes	904
2023	904
2020	905
Ejemplo de aplicaciones	906
Proceso de eventos de DynamoDB	906
Antes de empezar	906
Paso 1: Inicializar la aplicación	906
Paso 2: probar la aplicación localmente	907
Paso 3: crear el paquete de la aplicación	907
Paso 4: implementación de la aplicación	908
Sigüientes pasos	909
Procesar eventos de Amazon S3	909
Antes de empezar	909
Paso 1: Inicializar la aplicación	909
Paso 2: Crear el paquete de la aplicación	910
Paso 3: implementación de la aplicación	911
Paso 4: probar la aplicación localmente	911
Sigüientes pasos	912
Terraform compatibilidad	913
Introducción	913
Requisitos previos	913
Uso de comandos de la CLI de AWS SAM con Terraform	914
Configuración para proyectos de Terraform	914
Configuración para Terraform Cloud	920

Uso de la CLI de AWS SAM con Terraform	922
Pruebas locales con sam local invoke	922
Pruebas locales con sam local start-api	922
Pruebas locales con sam local start-lambda	924
Limitaciones de Terraform	925
Uso de la CLI de AWS SAM con Serverless.tf	925
Referencia de Terraform	926
AWS SAM referencia de funciones compatibles	926
Referencia específica de Terraform	926
metadatos de sam	926
Compatibilidad de Terraform con la CLI de AWS SAM	929
¿Qué es el? AWS SAMCLI	930
¿Cómo puedo usar la CLI de AWS SAM con Terraform?	931
Sigüientes pasos	931
AWS CDK soporte	932
Introducción	932
Requisitos previos	932
Crear y probar localmente una aplicación AWS CDK	933
Pruebas a nivel local	935
Ejemplo	936
Construcción	937
Ejemplo	937
Implementación	938
Publicar para que otros la usen	939
Requisitos previos	939
Publicación de una aplicación	941
Paso 1: Añadir una Metadata sección a la AWS SAM plantilla	941
Paso 2: crear el paquete de la aplicación	941
Paso 3: publicar la aplicación	942
Paso 4: compartir la aplicación (opcional)	942
Publicar una nueva versión de una aplicación existente	943
Temas adicionales	943
Propiedades de la sección metadatos	943
Propiedades	944
Casos de uso	946
Ejemplo	947

Historial de documentos	949
.....	cmlxxv

¿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 [AWS CloudFormation](#) recursos especializados sin servidor que se transforman en infraestructura durante la implementación. Este marco incluye dos componentes principales: el AWS SAM CLI y el proyecto. El AWS SAM proyecto es el directorio del proyecto de la aplicación que se crea cuando se ejecuta `sam init`. El AWS SAM proyecto incluye archivos como la AWS SAM plantilla, que incluye la especificación de la plantilla (la sintaxis abreviada que se utiliza para declarar los recursos).

Características principales

AWS SAM ofrece una variedad de beneficios que mejoran la experiencia del desarrollador, ya que le permiten:

Defina el código de infraestructura de su aplicación rápidamente, utilizando menos código

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

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

Use la CLI de AWS SAM para administrar su 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 [El AWS SAM CLI](#).

Aprovisione rápidamente los permisos entre los recursos con AWS SAM conectores

Use 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 [Para obtener más información, consulte administración de los permisos de recursos con conectores de AWS SAM ..](#)

Sincronice de forma continua los cambios locales en la nube a medida que vaya desarrollando contenido

Utilice el AWS SAM CLI `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, consulte [Introducción al uso sam sync de la sincronización 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, consulte [Compatibilidad de la CLI de AWS SAM con Terraform](#).

Información relacionada

- Para obtener información sobre cómo AWS SAM funciona, consulte [¿Cómo funciona AWS SAM](#).
- 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 funciona AWS SAM

AWS SAM consta de dos componentes principales que se utilizan para crear la aplicación sin servidor:

1. [El AWS SAM proyecto](#)— 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. Esta plantilla incluye la especificación de la AWS SAM plantilla: 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 su aplicación sin servidor.
2. [El AWS SAM CLI](#)— Una herramienta de línea de comandos que puede usar con su AWS SAM proyecto e integraciones de terceros compatibles para crear y ejecutar sus aplicaciones sin servidor. La AWS SAM CLI) es 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 SAM CLI para ejecutar los comandos en su AWS SAM proyecto, que es la forma de inicializar, compilar, probar e implementar su aplicación sin servidor.

¿Eres nuevo en la tecnología sin servidor?

Le recomendamos que haga una reseña [Conceptos sin servidor para AWS Serverless Application Model](#).

¿Cuál es la especificación AWS SAM de la 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:

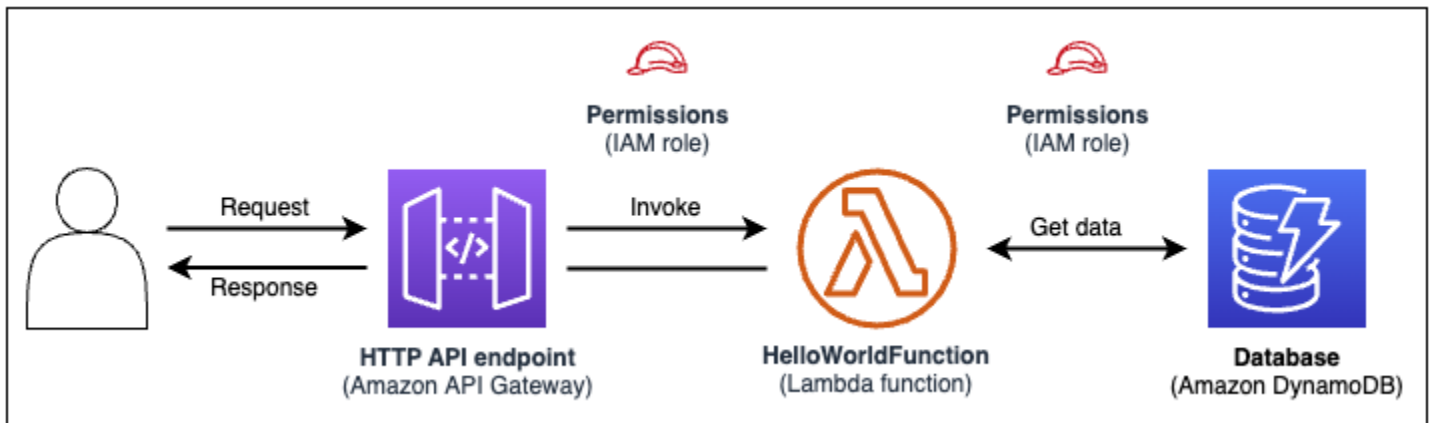
- Basado en AWS CloudFormation: la AWS CloudFormation sintaxis se utiliza directamente en la AWS SAM plantilla y se aprovecha su amplia compatibilidad con las configuraciones de recursos y propiedades. Si ya lo conoce AWS CloudFormation, no necesita aprender un nuevo servicio para administrar el código de infraestructura de aplicaciones.
- Una extensión de AWS CloudFormation: AWS SAM ofrece su propia sintaxis única que se centra específicamente en acelerar el desarrollo sin servidores. Puede usar la AWS SAM sintaxis AWS CloudFormation y la sintaxis en la misma plantilla.
- Una sintaxis abstracta y abreviada: con la sintaxis de AWS SAM , puede 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. AWS CloudFormation

¿Qué son el AWS SAM proyecto y la 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 AWS 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 ejecute 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: nodejs12.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 AWS CloudFormation. Durante la implementación, AWS SAM transforma las 23 líneas de código en la AWS CloudFormation sintaxis necesaria para generar estos recursos. AWS;La AWS CloudFormation plantilla transformada contiene más de 200 líneas de código!

AWS CloudFormation Plantilla transformada

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Resources": {
    "getAllItemsFunction": {
      "Type": "AWS::Lambda::Function",
      "Metadata": {
        "SamResourceId": "getAllItemsFunction"
      },
      "Properties": {
        "Code": {
          "S3Bucket": "aws-sam-cli-managed-default-samclisourcebucket-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 utilizarla con AWS SAM, se definen 23 líneas de código de infraestructura. AWS SAM transforma el código en las más de 200 líneas de código de AWS CloudFormation necesarias para aprovisionar la aplicación.

¿Qué es el? AWS SAMCLI

AWS SAMCLI es una herramienta de línea de comandos que puede utilizarse con las plantillas de AWS SAM 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 su aplicación para su implementación.
- Realizar pruebas y depuraciones locales.
- Implemente la aplicación.
- Configurar las canalizaciones de implementación de CI/CD.
- Supervisar y solucionar los problemas de su aplicación en la nube.
- Sincronizar los cambios locales con la nube a medida que vaya desarrollándose.
- Y mucho más.

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

Inicio de un nuevo proyecto

Seleccione una de las plantillas iniciales o elija una ubicación de plantilla personalizada para comenzar un nuevo proyecto.

A continuación se muestra el comando `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, consulte [Crea tu aplicación en AWS SAM](#).

Compilación de su 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 `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, consulte [Cree su aplicación](#).

Realización de pruebas y depuraciones locales

En la máquina local, simule eventos, pruebe las API, invoque funciones y mucho más para depurar y probar su 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 usar 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, consulte [Pruebe su aplicación](#) y [Depure su 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 AWS CloudFormation, transforma nuestra plantilla en ella y la implementa AWS 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 [Implemente su aplicación y sus recursos](#).

Configuración de las canalizaciones de implementación de CI/CD

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

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

[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, consulte [Implemente con sistemas y canalizaciones de CI/CD](#).

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

Consulte información importante sobre los recursos desplegados, recopile registros y utilice 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, consulte [Supervise su aplicación](#).

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

A medida que vaya desarrollando en su máquina local, sincronice automáticamente los cambios con la nube. Consulte rápidamente los cambios y realice 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-15GLOUR9LMT1W

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

Invoque y transfiera eventos a los recursos compatibles en la nube.

En este caso, utilizamos el comando `aws 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.

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, consulte [¿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 obtener más información, consulte La [infraestructura como código](#) en la introducción a un documento DevOps AWS AWS técnico.

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, consulte 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](#).

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, estará listo para hacerlo [Instale la CLI AWS SAM](#), lo que puede hacer en los sistemas operativos Linux, Windows y macOS. Una vez completada la instalación, si lo desea, puede seguir el tutorial de AWS SAM Hello World. Al seguir este tutorial, se explicará el proceso de creación de una aplicación básica sin servidor con AWS SAM. Tras completar el tutorial, estará listo para revisar los conceptos detallados en [Cómo usar AWS Serverless Application Model \(AWS SAM\)](#).

Temas

- [AWS SAM requisitos previos](#)
- [Instale la CLI AWS SAM](#)
- [Tutorial: Implemente una aplicación Hello World con AWS SAM](#)

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

- [Paso 1: Crear una AWS cuenta](#)
- [Paso 2: Crear una cuenta de usuario de IAM](#)
- [Paso 3: Crear un ID de clave de acceso y la clave de acceso secreta](#)
- [Paso 4: Instale el AWS CLI](#)
- [Paso 5: utilícelo AWS CLI para configurar AWS las credenciales](#)

- [Sigüientes pasos](#)

Paso 1: Crear una AWS cuenta

Si no tiene una Cuenta de AWS, complete los siguientes pasos para crear una.

Para suscribirse a una Cuenta de AWS

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

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

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

Paso 2: Crear una cuenta de usuario de IAM

Para crear un usuario administrador, elija una de las siguientes opciones.

Elegir una forma de administrar el administrador	Para	Haga esto	También puede
En IAM Identity Center (recomendado)	Usar credenciales a corto plazo para acceder a AWS. Esto se ajusta a las prácticas	Siga las instrucciones en Introducción en la Guía del usuario de AWS IAM Identity Center .	Configure el acceso programático configurando el AWS CLI que se utilizará AWS IAM Identity Center en la

Elegir una forma de administrar el administrador	Para	Haga esto	También puede
	recomendadas de seguridad. Para obtener información sobre las prácticas recomendadas, consulte Prácticas recomendadas de seguridad en IAM en la Guía del usuario de IAM.		Guía del AWS Command Line Interface usuario.
En IAM (no recomendado)	Usar credenciales a largo plazo para acceder a AWS.	Siga las instrucciones en Creación del primer grupo de usuarios y usuario de administrador de IAM en la Guía del usuario de IAM.	Configurar el acceso programático mediante Administración de las claves de acceso de los usuarios de IAM en la Guía del usuario de IAM.

Paso 3: Crear un ID de clave de acceso y la clave de acceso secreta

Para acceder a la CLI, necesita un ID de clave de acceso y una clave de acceso secreta. Cuando sea posible, utilice credenciales temporales en lugar de claves de acceso. Las credenciales temporales incluyen un ID de clave de acceso y una clave de acceso secreta, pero, además, incluyen un token de seguridad que indica cuándo caducan las credenciales. Para obtener más información, consulte [Uso de credenciales temporales con AWS recursos](#) en la Guía del usuario de IAM.

Los usuarios necesitan acceso programático si quieren interactuar con personas AWS ajenas a AWS Management Console. La forma de conceder el acceso programático depende del tipo de usuario que acceda. AWS

Para conceder acceso programático a los usuarios, elija una de las siguientes opciones.

¿Qué usuario necesita acceso programático?	Para	Mediante
Identidad del personal (Usuarios administrados en el IAM Identity Center)	Usa credenciales temporales para firmar las solicitudes programáticas a los AWS CLI, AWS SDK o las API. AWS	Siga las instrucciones de la interfaz que desea utilizar: <ul style="list-style-type: none"> • Para ello AWS CLI, consulte Configuración del uso AWS IAM Identity Center en AWS CLI la Guía del AWS Command Line Interface usuario. • Para ver AWS los SDK, las herramientas y las AWS API, consulte la autenticación del IAM Identity Center en la Guía de referencia de AWS los SDK y las herramientas.
IAM	Utilice credenciales temporales para firmar las solicitudes programáticas a los AWS SDK o las AWS CLI API. AWS	Siga las instrucciones de Uso de credenciales temporales con AWS recursos de la Guía del usuario de IAM.
IAM	(No recomendado) Utilice credenciales de larga duración para firmar las solicitudes programáticas a los AWS CLI, AWS SDK o las API. AWS	Siga las instrucciones de la interfaz que desea utilizar: <ul style="list-style-type: none"> • Para ello AWS CLI, consulte Autenticación con credenciales de usuario de IAM en

¿Qué usuario necesita acceso programático?	Para	Mediante
		<p data-bbox="1101 260 1451 338">la Guía del usuario.AWS Command Line Interface</p> <ul data-bbox="1068 365 1487 1024" style="list-style-type: none"> <li data-bbox="1068 365 1487 730">• Para obtener información AWS sobre los SDK y las herramientas, consulte Autenticarse con credenciales de larga duración en la Guía de referencia de los AWS SDK y las herramientas. <li data-bbox="1068 751 1487 1024">• Para obtener información AWS sobre las API, consulte Administrar las claves de acceso para los usuarios de IAM en la Guía del usuario de IAM.

Paso 4: 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 5: utilícelo AWS CLI para configurar AWS las credenciales

Para configurar las credenciales con AWS CLI

1. Ejecute el comando `aws configure` desde la línea de comandos.
2. Configure 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 explícitamente el perfil que se va a utilizar. Para obtener más información sobre el archivo `credentials`, consulte [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, consulte [Quick setup](#) en la Guía del usuario de AWS Command Line Interface .

Siguientes pasos

Ahora está listo para instalar AWS SAMCLI y empezar a usarlo AWS SAM. Para instalar el AWS SAMCLI, consulte [Instale la CLI AWS SAM](#).

Instale la CLI AWS SAM

Instale la última versión de la interfaz de línea de AWS Serverless Application Model comandos (AWS SAMCLI) en los sistemas operativos compatibles.

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 la CLI de AWS SAM](#).

¿Es la primera vez que instala AWS SAM CLI?

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

1. Registrarse para obtener una AWS cuenta.
2. Creación de un usuario de IAM no administrativo.
3. Creación de un identificador de clave de acceso y de una clave de acceso secreta.
4. Instalación del AWS CLI.
5. Configuración de AWS credenciales.

Temas

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

Instalación de la CLI de AWS SAM

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, puede utilizar el instalador gestionado por la comunidad. Para obtener más información, consulte [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, consulte [Cambiar de Homebrew](#).

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

Linux

arm64 - command line installer

1. Descargue el [archivo .zip de AWS SAM CLI](#) a un directorio de su elección.
2. (Opcional) Puede comprobar la integridad del instalador antes de la instalación. Para ver instrucciones, consulte [Opcional: compruebe la integridad del AWS SAMCLI instalador](#).
3. Descomprima los archivos de instalación en el directorio que prefiera. A continuación, se muestra un ejemplo de uso del subdirectorio `sam-installation`.

Note

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

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

4. Instale la CLI de AWS SAM 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. Verifique la instalación.

```
$ sam --version
```


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

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

x86_64 - command line installer

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

- (Opcional) Puede comprobar la integridad del instalador antes de la instalación. Para ver instrucciones, consulte [Opcional: compruebe la integridad del AWS SAMCLI instalador](#).
- Descomprima los archivos de instalación en el directorio que prefiera. A continuación, se muestra un ejemplo de uso del subdirectorio `sam-installation`.

 Note

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

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

- Instale la CLI de AWS SAM 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
```

- Verifique 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>
```

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.

GUI - All users

Para descargar el instalador del paquete e instalar el AWS SAMCLI

Note

Si instaló previamente la CLI de AWS SAM mediante Homebrew o pip, primero debe desinstalarlo. Para ver instrucciones, consulte [Desinstalación de la CLI de AWS SAM](#).

1. Descarga macOS pkg en el directorio que prefieras:

- Para ordenadores 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 verificar la integridad del instalador antes de la instalación. Para ver instrucciones, consulte [Opcional: compruebe la integridad del AWS SAMCLI instalador](#).

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

Note

Para invocar la CLI de AWS SAM 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 elija.

5. El AWS SAM y la CLI procederán a la instalación y mostrarán “La instalación se ha realizado correctamente”. Pulse Cerrar.

Para comprobar que la instalación se ha realizado correctamente

- Compruebe que la CLI de AWS SAM 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

Para descargar e instalar el AWS SAMCLI

Note

Si instaló previamente la CLI de AWS SAM mediante Homebrew o pip, primero debe desinstalarlo. Para ver instrucciones, consulte [Desinstalación de la CLI de AWS SAM](#).

1. Descarga macOS pkg en el directorio que prefieras:

- Para ordenadores 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 verificar la integridad del instalador antes de la instalación. Para ver instrucciones, consulte [Opcional: compruebe la integridad del AWS SAMCLI instalador](#).

2. Ejecute el archivo descargado y siga las instrucciones que aparecen en pantalla para continuar con los pasos de introducción, léame y licencia.
3. En Destination Select, selecciona Instalar solo para mí. Si no ve esta opción, vaya 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`. Seleccione una ubicación para la que tenga permisos de escritura. Para cambiar la ubicación de instalación, seleccione local y elija su ubicación. Pulse Continuar cuando haya terminado.
2. Si en el paso anterior no tenía la opción de elegir Instalar solo para mí, seleccione Cambiar ubicación de instalación > Instalar solo para mí y pulse Continuar.
3. Pulse Instalar.
5. El AWS SAM y la CLI procederán a la instalación y mostrarán “La instalación se ha realizado correctamente”. Pulse Cerrar.

Para crear un enlace simbólico

- Para invocar la CLI de AWS SAM con el comando `sam`, debe crear manualmente un enlace simbólico entre la CLI de AWS SAM y su `$PATH`. Cree su 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 `$PATH`, `sudo` no es obligatorio. De lo contrario, se requiere el `sudo`.
- **path-to**: ruta al lugar donde instaló la CLI de AWS SAM. Por ejemplo, `/Users/myUser/Desktop`.
- **path-to-symlink-directory**— Su variable de `$PATH` entorno. La ubicación predeterminada es `/usr/local/bin`.

Para comprobar que la instalación se ha realizado correctamente

- Compruebe que la CLI de AWS SAM 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

Para descargar e instalar el AWS SAMCLI

Note

Si instaló previamente la CLI de AWS SAM mediante Homebrew o pip, primero debe desinstalarlo. Para ver instrucciones, consulte [Desinstalación de la CLI de AWS SAM](#).

1. Descarga macOS pkg en el directorio que prefieras:

- Para ordenadores 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 verificar la integridad del instalador antes de la instalación. Para ver instrucciones, consulte [Opcional: compruebe la integridad del AWS SAMCLI instalador](#).

2. Modifique y ejecute 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

- Compruebe que la CLI de AWS SAM 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

Para descargar e instalar el AWS SAMCLI

 Note

Si instaló previamente la CLI de AWS SAM mediante Homebrew o pip, primero debe desinstalarlo. Para ver instrucciones, consulte [Desinstalación de la CLI de AWS SAM](#).

1. Descarga macOS pkg en el directorio que prefieras:
 - Para ordenadores 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 verificar la integridad del instalador antes de la instalación. Para ver instrucciones, consulte [Opcional: compruebe la integridad del AWS SAMCLI instalador](#).

2. Defina un directorio de instalación en el que tenga 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. Guarde el archivo xml y compruebe que es válido ejecutando lo siguiente:

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

El resultado debe mostrar las preferencias que se aplicarán a la CLI de AWS SAM.

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 la CLI de AWS SAM con el comando `sam`, debe crear manualmente un enlace simbólico entre la CLI de AWS SAM y su `$PATH`. Cree su 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 `$PATH`, `sudo` no es obligatorio. De lo contrario, se requiere el `sudo`.
- **`path-to`**: ruta al lugar donde instaló la CLI de AWS SAM. Por ejemplo, `/Users/myUser/Desktop`.
- **`path-to-symlink-directory`**— Su variable de `$PATH` entorno. La ubicación predeterminada es `/usr/local/bin`.

Para comprobar que la instalación se ha realizado correctamente

- Compruebe que la CLI de AWS SAM 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. Descargue la CLI de AWS SAM de [64 bits](#).

 Note

Si utiliza una versión de 32 bits de Windows, consulte [Instalación de la CLI de AWS SAM en Windows 32 bits](#).

- (Opcional) Puede comprobar la integridad del instalador antes de la instalación. Para ver instrucciones, consulte [Opcional: compruebe la integridad del AWS SAMCLI instalador](#).
- Verifique la instalación.

Tras completar la instalación, verifíquela 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>
```

- 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 `sam init` debido a las `MAX_PATH` limitaciones de Windows 10. Para resolver este problema, debe configurar el comportamiento de las nuevas rutas de acceso largo.

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

- Instale Git

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

Solución de errores de la instalación de

Linux

Error de Docker: “No se puede conectar al daemon de Docker. ¿Se está ejecutando el daemon de Docker en este host?”

En algunos casos, es posible que tenga que reiniciar su instancia para que el `ec2-user` tenga los permisos necesarios para acceder al daemon de Docker. Si recibe este error, intente reiniciar la instancia.

Error del intérprete de comandos: “no se ha encontrado el comando”

Si recibe este error, su intérprete de comandos no puede localizar el ejecutable AWS SAMCLI en la ruta. Compruebe la ubicación del directorio en el que instaló el ejecutable AWS SAMCLI y, a continuación, compruebe que el directorio se encuentra en su ruta.

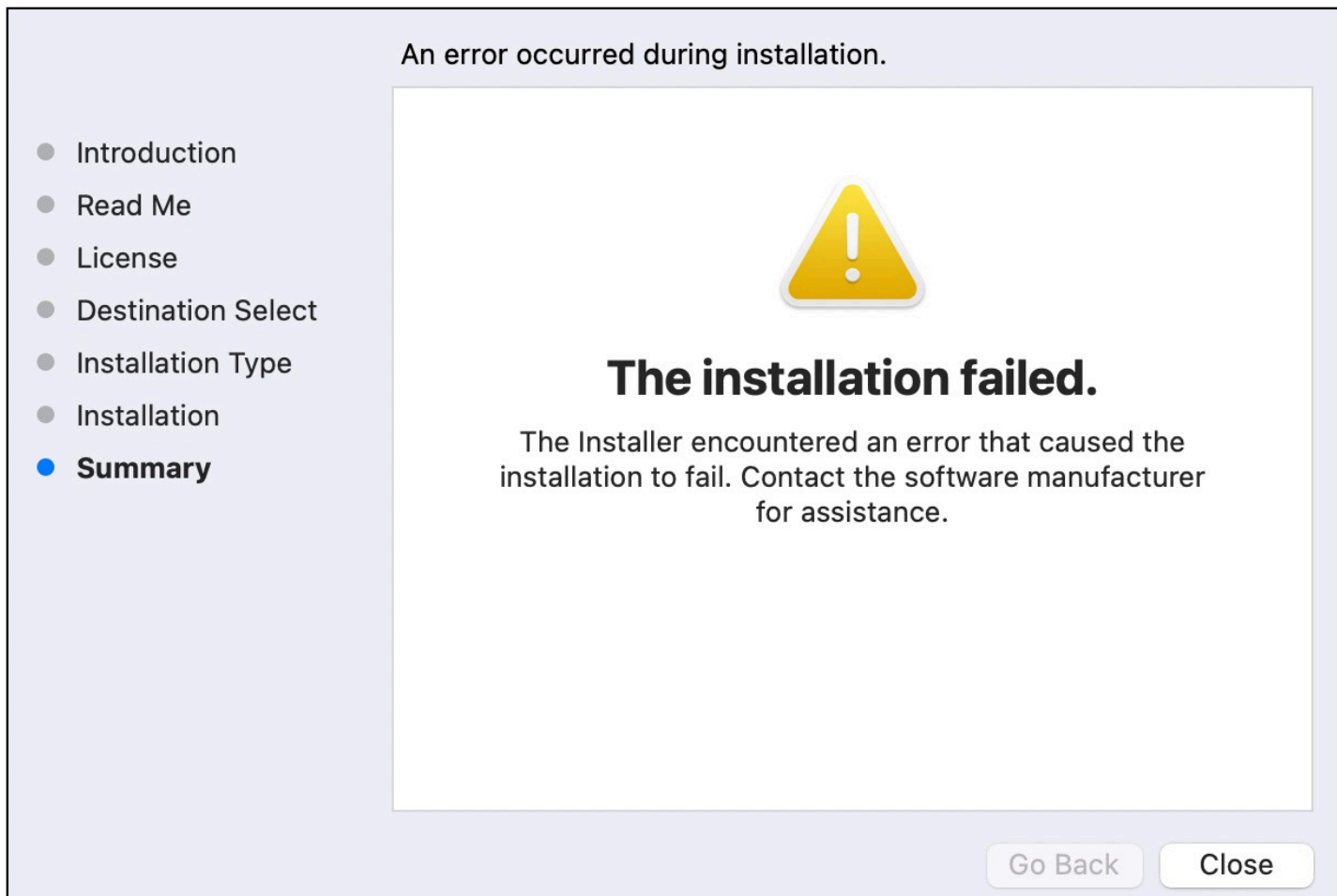
AWS SAMCLlerror: «/lib64/libc.so.6: no se ha encontrado la versión `GLIBC_2.14' (requerida por /usr/local/dist/libz.so.1)» aws-sam-cli

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

- Actualice 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 [Instale la CLI AWS SAM](#).

macOS

Error en la instalación



Si está instalando la AWS SAMCLI para su usuario y ha seleccionado un directorio de instalación para el que no tiene permisos de escritura, podría producirse este error. Haga una de estas dos operaciones:

1. Seleccione otro directorio de instalación para el que tenga permisos de escritura.
2. Elimine el instalador. A continuación, descárguelo y ejecútelo de nuevo.

Siguientes pasos

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

- [Tutorial: Implemente una aplicación Hello World con AWS SAM](#)— tep-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 SAMCLI instalador

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:

- Verifique la firma del archivo instalador.
- Compruebe el valor hash del instalador del paquete.

Cuando esté disponible para su plataforma, le 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

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

Verifique 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. Utilice la clave pública principal para verificar la clave pública del firmante.

2. Utilice la clave pública del firmante para verificar el instalador del paquete AWS SAMCLI.

Para verificar la integridad de la clave pública del firmante

1. Copie la clave pública principal y guárdela 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
D9jE0nvupx4tvvhfBHz5EcUHCE0d14MTqdBy6vVAshozgxVb9RE8JpECn5lw7XC69
4Y7Gy1TKKQMEWtDXElkGxIFdUwvWjSnPlzfnoXwQYGeE93CUS3h5dImP22Yk1Ct6
eGGhlcbg1X4L8EpFMj7GvcsU8f7ziVI/PyC1Xwy39Q8/I67ip5eU5ddx0/xHqrbL
YC7+8pJPbRMej2twT2LrcpWYAbrMtRoa6WfE0/thoo3xhHpIMHdPFAA86ZNGIN
kRLjGUg7jnPTRW40in3pCc8nT4Tfc1QERkHm641gTC/jUvpmQsM6h/FUVP2i5iE/
JHpJcMuL2Mg6zDo3x+3gTCf+Wqz3rZzxB+wQT3yryZs6efcQy7nR0iRxYBxCSXX0
2cNYzsYlb/bYaW8yqWIHD5IqKhW269gp2E5Khs60zgs3CorMb5/xHgXjUCVgcu8a
a8ncdf9fj13WS5p0ohetPb0Z2jWv+MaqrZ0mUIgKbA4RpWZ/fU97P5BW9ylwmIDB
sWy0cMxg8M1vSdLytPieogaM0qMg3u5qXRGBr6Wmevkty0qgnmpGGc5zPiUbtOE8
CnFFqyxBpj5IOng0KZGVihvn+iRrxv6G07WW092+Dc6m94U0EEiBR7Qi0wARAQAB
tDRBV1MgU0FNIENMSSBQcm1tYXJ5IDxhd3Mtc2FtLWNsaS1wcm1tYXJ5QGFTYXpv
bi5jb20+iQI/BBMBCQApBQJkbksZAhsvBQkHhM4ABwsJCAcDAgEGFQgCCQoLBBYC
AwECHgECF4AACgkQQv1fen0tiFqTuhAAzi5+ju5UV0WqHkev0JS008T4QB8HcqAE
SV03mY6/j29knkcL8ubZP/DbpV7QpHPI2PB5qSXsiDTP3IYPbeY78zHSDjljaIK3
njJLMScFeGPYfPpwMsuY4nzrRIgAtXShPA8N/k4ZJcafnpNqKj7QnPxIC1KaIQWm
p0tvb8msUF3/s0UTa5Ys/1NRhVC0eGg32ogXGdojZA2kHZWdm9udLo4CDrDcrQT7
NtDcJASapXSQL63XfAS3snEc4e1941YxcjFYZ33rel8K9juyDZfi1s1WR/L3AviI
QFIaqSHzy0tP1oinUkoVwL8ThevKD3Ag9CZf1ZLzNCV7yq1F8R1hEZ4zce/3s9E1
WzCFsozb5HfE1AZonmrDh3Sy0EIBMCS6vG5dWnvJrAuSYv2rX38++K5Pr/MIAfOX
DOI1rtA+XDshNv91SwSy0lt+iClawZAN09IXCiN1r0YcVQlwzDFwCNWDgkwd0qS0
g0A2f8NF91E5nBbeEuYquo011Vy8+ICbg0F9LoWZlnVh7/RyY6ssowiU9vGUNHI
L8f9jqRspIz/Fm3JD86ntZxLVGkeZUz62FqErdohYfkFIVcv7GONTEyrz5HL1npv
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. Copie la clave pública del firmante y guárdela 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)

mQINBGRtS20BEAC7GjaAwverrB1zNEu2q3EGI6HC37WzwL5dy30f4LirZ0WS3piK
oKfTqPjXPrlCf1GL2mMqUSgSnpEbPNXuvWTW1CfSnnjwuH8ZqbvvUQyHJwYpKw
KMwb+8V0bzzQkMzDVqoLYQCi5XyGpAuo3wroxXSzG6r/mIhbiq3aRnL+2lo4X0Yk
r7q9bhBqbJhzjkm7N62PhPWmi/+EGdEBakA1pReE+cKjP2UAp5L6CPSHQ12fRKL
9BumitNfFHHs1JJZgZSCCruiWny3XkUaXUEMfyoE9nNbfqNvuqV2KjWguZCXASgz2
ZSPF4DTVIBMfP+xrZGQSWdGU/67QdysDQW81TbF0jK9ZsRwwGC4kbg/K98IsCNHT
ril5RZbyr8pw3fw7jYjjI2E1AacRwP53iRzvutm5AruPpLfoKDQ/tKzBUYItBwlu
Z/diKgcqtW7xDlyqNyTN8xPFpM02I8IsZ2Pd1131htdFiZMiin1RQG9pV9p2vHS
eQVY2uKcNvnA6vFCQYKXP7p0IwReuPNzDvECUsidw8VTakTqZsANT/bU17e4KuKn
+JgbNrk0asJX37sDb/9ruysozLvy78ozYKJDLmC3yoRQ8DhEjviT4cnjORgNmvnZ
0a5AA/DJPPQW4buRrXdxu+fITzBxQn2+G0/iDNCxtJaq5SYVBKjTmTWPUJwARAQAB
tDBBV1MgU0FNIENSSBUZWFtIDxhd3MtZ2FtLWNsaS1zaWduZXJAYW1hem9uLmNv
bT6JAj8EEwEJACKFAMrtS20CGy8FCQPCZwAHCwkIBwMCAQYVCAIJCgsEFgIDAQIe
AQIXgAAKCRDHoF9D/grd+1E4D/4kJW65He2LNsBLTta7lcGfsEXCf4zgIvkytS7U
3R36zMD8IEyWJjLZ+aPkIP8/jFjrF14pVhBU7vX85Iut1vV7m+8BgWt25mJhnoJ9
KPjXGra9mYP+Cj8zFACjvt13NBAPodyfCfCTWsU3umF9ArOFICcrGCzHX2SS7wX5
h9n0vYRZxk5Qj5FsgskKAQLq33CKFAMlaqZnL5gWRvTeycSIxsysus+stX+8YBPC0
J64f7+y+MPIP1+m2nj1VXg1xLEMMVa08oWcc0MiakgzDev3LCrPy+wdwdn7Ut7oA
pna3DNy9aYnd2lh6vUCJeJ+Yi1B12jYpzLcCLKrHUmIn9/rRSz70rbg8P181kfPu
G/M7CD5FwhxP3p4+0XoGwxQefrV2jqPsnBLae7xbYJiJAhpjWDQhuNGUbPcDmqk
aH0Q3XU8AonJ8YqaQ/q3VZ3JBih3TbBr0Xsvd59cwxYyf83aJ/WLCb2P8y75zDad
ln0P713ThF5J/Afj9Hj09waFV0Z2W2ZZe4rU20JTAiXEtM8xsFMrc7TCUacJtJGs
u4kdBmXREcVpSz65h9ImSy2ner9qktnVVCW4mZPj63IhB37YtoLAMyz3a3R2RFNk
viEX8fo0TUg1FmWofTxZ9P91QwLoTajkDrh26ueIe45sG6Uxua2AP4Vo37cFfCj
ryV80okCHAQQAQkABgUCZG5MWAACKRBC/V96c62IWmg1D/9idU43kW8Zy8Af1j81

```

```
Am3lI4d9ks0leeKRZqxo/SZ5rovF32D02nw7XRXq1+EbhgJaI3Qww0i0U0pfAMVT
4b9TdxH+n+tzqCHh3jZqmo9sw+c9WfXYJN1hU9bLzCHXs8h0TbyoE2EuXx56ds9
L/BWCcd+LIvawp01ggFfavVx/QF4C7nBKjnJ66+xxwfgVIKR7oG1qDiHMfp9ZWh5
HhEqZo/nrNhdY0h3sczEdqC2N6eIa8mgHffHZdKudDMXIXHbgdhW9pcZXDIktVf7
j9wehsW0yYXiRgR0dz7DI26AUG4JLh5FTtx9XuSBdEsI69Jd4dJuibmgtImzbZjn
7un8DJWIyqi7Ckk96Tr4oXB9mYAXaWLR4C9j5XJhMNZgk0ycuY2DADnbGmSb+1kA
ju77H4ff84+vMDwUzUt2Wwb+GjzXu2g6Wh+bWhGSirYle1+6xYrI6beu1BDCFLq+
VZFE8WggjJHpwL7CiqadfVIQaw4HY0jQFTSdwzPWhJvYjXF0hMkyCcjsbBtmB+z
/otfgySyQqThrD48RWS5GuyqCA+pK3UNmEJ11c1AXMdTn2VWInR1N0JNALQ2du3y
q8t1vMsErV0J7pkZ50F4ef17PE6DKrXX8ilwGFyVuX5ddyT/t9J5pC3sRwHWXVZx
GXwoX75FwIEHA3n5Q7rZ69Ea6Q==
=ZI07
-----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 FE0ADDFA: 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, *FE0ADDFA*.

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

```
$ gpg --fingerprint FE0ADDFA
```

```
pub 4096R/FE0ADDFA 2023-05-23 [expires: 2025-05-22]
Key fingerprint = 37D8 BE16 0355 2DA7 BD6A 04D8 C7A0 5F43 FE0A DDFA
uid AWS SAM CLI Team <aws-sam-cli-signer@amazon.com>
```

La huella digital debe coincidir con la siguiente.

```
37D8 BE16 0355 2DA7 BD6A 04D8 C7A0 5F43 FE0A DDFA
```

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

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

```
$ gpg --check-sigs FE0ADDFA
```

```
pub 4096R/FE0ADDFA 2023-05-23 [expires: 2025-05-22]
uid                               AWS SAM CLI Team <aws-sam-cli-signer@amazon.com>
sig!3    FE0ADDFA 2023-05-23  AWS SAM CLI Team <aws-sam-cli-signer@amazon.com>
sig!      73AD885A 2023-05-24  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 comprobar la integridad del instalador del paquete de la CLI de AWS SAM

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. Verifique 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 Tue 30 May 2023 10:03:57 AM UTC using RSA key ID FE0ADDFA
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: 37D8 BE16 0355 2DA7 BD6A 04D8 C7A0 5F43 FE0A DDFA
```

- 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, consulte [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 aws-sam-cli GitHub repositorio 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. Utilice la clave pública principal para verificar la clave pública del firmante.
2. Utilice la clave pública del firmante para verificar el instalador del paquete AWS SAMCLI.

Para verificar la integridad de la clave pública del firmante

1. Copie la clave pública principal y guárdela 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  
4Y7Gy1TKKQMEWtDXE1kGxIFdUwWjSnPlzfnoXwQYGeE93CUS3h5dImP22Yk1Ct6  
eGGh1cbg1X4L8EpFMj7GvcsU8f7ziVI/PyC1Xwy39Q8/I67ip5eU5ddx0/xHqrbL  
YC7+8pJPbRMej2twT2LrCpWYAbprMtRoa6WfE0/thoo3xhHpIMHdPFAA86ZNGIN  
kRLjGUg7jnPTRW40in3pCc8nT4Tfc1QERkHm641gTC/jUvpmQsM6h/FUVP2i5iE/  
JHpJcMuL2Mg6zDo3x+3gTCf+Wqz3rZzxB+wQT3yryZs6efcQy7nR0iRxYBxCSXX0  
2cNYzsYLb/bYaW8yqWIHD5IqKhW269gp2E5Khs60zgS3CoRmb5/xHgXjUCVgcu8a
```

```
a8ncdf9fj13WS5p0ohetPb02ZjWv+MaqrZ0mUIgKbA4RpWZ/fU97P5BW9y1wmIDB
sWy0cMxg8MlvSdLytPieogaM0qMg3u5qXRGBr6Wmevkty0qgnmpGGc5zPiUbt0E8
CnFFqyxBpj5I0nG0KZGVihvn+iRrxrv6G07WW092+Dc6m94U0EEiBR7Qi0wARAQAB
tDRBV1MgU0FNIENSSSBQcmLtYXJ5IDxhd3Mtc2FtLWNsaS1wcmLtYXJ5QGFTYXpv
bi5jb20+iQI/BBMBCQApBQJkbszAhsvBQkHhM4ABwsJCAcDAgEGFQgCCQoLBBYC
AwEChgECF4AACgkQQv1fen0tiFqTuhAAzi5+ju5UV0WqHkev0JS008T4QB8HcqAE
SV03mY6/j29knkcL8ubZP/DbpV7QpHPi2PB5qSXsiDTP3IYPbeY78zHSDjljaIK3
njJLMScFeGPyfPpwMsuY4nzrRIgAtXShPA8N/k4ZJcafnpNqKj7QnPxIC1KaIQWm
p0tvb8msUF3/s0UTa5Ys/1NRhVC0eGg32ogXGdojZA2kHZWdm9udLo4CDrDcrQT7
NtDcJASapXSQL63XfAS3snEc4e1941YxcjFYZ33rel8K9juyDZfi1sLWR/L3AviI
QFIaqSHzy0tP1oinUkoVwL8ThevKD3Ag9CZf1ZLzNCV7yq1F8R1hEZ4zcE/3s9E1
WzCFsozb5HfE1AZonmrDh3Sy0EIBMCS6vG5dWnvJrAuSYv2rX38++K5Pr/MIAf0X
D0I1rtA+XDshNv91SwSy01t+iClawZAN09IXCiN1r0YcVQ1wzDFwCNWDgkwd0qS0
g0A2f8NF91E5nBbeEuYquo011Vy8+ICbg0Fs9LoWZlnVh7/RyY6ssowiU9vGUnHI
L8f9jqRspIz/Fm3JD86ntZxLVGkeZUz62FqErdohYfkFIVcv7GONTEyrz5HL1npv
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. Copie la clave pública del firmante y guárdela 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)

mQINBGRtS20BEAC7GjaAwverrB1zNEu2q3EGi6HC37Wzwl5dy30f4LirZ0WS3piK
```

```

oKfTqPjXPrlCf1GL2mMqUSgSnpEbPNXuvWTW1CfSnnjwuH8ZqbvVUQyHJwQyYpKm
KMwb+8V0bzzQkMzDVqoLYQCi5XyGpAuo3wroxXSzG6r/mIhbiq3aRnL+2lo4X0Yk
r7q9bhBqbJhzjkm7N62PhPWmi/+EGdEBakA1pReE+cKjP2UAp5L6CPSHQ12fRKL
9BumitNfFHHs1JZgZSCCruiWny3XkUaXUEMfyoE9nNbfqNvuqV2KjWguZCXASgz2
ZSPF4DTVIBMfP+xrZGQSwdGU/67QdysDQW81TbF0jK9ZsRwwGC4kbg/K98IsCNHT
ril5RZbyr8pw3fw7jYjjI2E1AacRWp53iRzvutm5AruPpLfoKDQ/tKzBUYItBwLu
Z/diKgcqtW7xDlyqNyTN8xFPFqM02I8IsZ2Pd1131htdFiZMiin1RQG9pV9p2vHS
eQVY2uKcNvnA6vFCQYKXP7p0IwReuPNzDvECUsidw8VTakTqZsANT/bU17e4KuKn
+JgbNrK0asJX37sDb/9ruysozLvY78ozYKJDLmC3yoRQ8DhEjviT4cnjORgNmvnZ
0a5AA/DJPPQW4buRrXdxu+fITzBxQn2+G0/iDNCxtJaq5SYVBKjTmTWPUJwARAQAB
tDBBV1MgU0FNIENSSBUZWFtIDxhd3Mtc2FtLWNsaS1zaWduZXJAYW1hem9uLmNv
bT6JAj8EEwEJACKFAMrT520CGy8FCQPCZwAHCwkIBwMCAQYVCAIJCgsEFgIDAQIe
AQIXgAAKCRDHoF9D/grd+1E4D/4kJW65He2LNsBLTta71cGfsEXCf4zgIvkytS7U
3R36zMD8IEyWJj1Z+aPkIP8/jFJrF14pVhBU7vX85Iut1vV7m+8BgWt25mJhnoJ9
KPjXGra9mYP+Cj8zFACjvt13NBAPodyfcfCTWsU3umF9Ar0FICcrGCzHX2SS7wX5
h9n0vYRZxk5Qj5FsgskKAQLq33CKFAMlaqZnL5gWRvTeycSIxsys+stX+8YBPC0
J64f7+y+MPIP1+m2nj1VXg1xLEMMVa08oWcc0MiakgzDev3LCrPy+wdwdn7Ut7oA
pna3DNy9aYNd2lh6vUCJeJ+Yi1B12jYpzLcCLKrHUmLn9/rRSz70rbg8P181kfPu
G/M7CD5FwhxP3p4+0XoGwxQefrV2jqpSnbLae7xbYJiJAhbpiWDQhuNGUbPcDmqk
aH0Q3XU8AonJ8YqaQ/q3VZ3JBih3TbBr0Xsvd59cwxYyf83aJ/WLCb2P8y75zDad
ln0P713ThF5J/Afj9Hj09waFV0Z2W2ZZe4rU20JTAiXEtM8xsFMrc7TCUacJtJGs
u4kdBmXREcVpSz65h9ImSy2ner9qktnVVCW4mZPj63IhB37YtoLAMyz3a3R2RFNk
viEX8fo0TUg1FmwHoftxZ9P91QwLoTajkDrh26ueIe45sG6Uxua2AP4Vo37cFfCj
ryV80okCHAQAQkABgUCZG5MWAACKRBC/V96c62IWmg1D/9idU43kW8Zy8Af1j81
Am31I4d9ks0leeKRZqxo/SZ5rovF32D02nw7XRXq1+EbhgJaI3Qww0i0U0pfAMVT
4b9TdxH+n+tzqCHh3jZqmo9sw+c9WFXyJN1hU9bLzCHXS8h0TbyoE2EuXx56ds9
L/BWCcd+LIvawp01ggFfavVx/QF4C7nBKjnJ66+xxwfgVIKR7oG1qDiHMfp9ZWh5
HhEqZo/nrNhdy0h3sczEdqC2N6eIa8mgHffHZdKudDMXIXHbgdhW9pcZXDiktVf7
j9wehsW0yYXiRgR0dz7DI26AUG4JLh5FTtx9XuSBdEsI69Jd4dJuibmgtImzbZjn
7un8DJWIyqi7Ckk96Tr4oXB9mYAXaWLR4C9j5XJhMNZgk0ycuY2DADnbGmSb+1kA
ju77H4ff84+vMDwUzUt2Wwb+GjzXu2g6Wh+bWhGSirYle1+6xYrI6beu1BDCFLq+
VZFE8WggjJHpwcl7CiqadfVIQaw4HY0jQFTSdwzPWhJvYjXF0hMkyCcjsstbMB+z
/otfgySyQqThrD48RWS5GuyqCA+pK3UNmEJ11c1AXMdTn2VWInR1NOJNALQ2du3y
q8t1vMsErV0J7pkZ50F4ef17PE6DKrXX8ilwGFyVuX5ddyT/t9J5pC3sRwHWXVZx
GXwoX75FwIEHA3n5Q7rZ69Ea6Q==
=ZI07
-----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 FE0ADDDFA: 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, **FE0ADDFA**.

- Utilice el valor de la clave para obtener y verificar la huella digital de la clave pública del firmante.

```
$ gpg --fingerprint FE0ADDFA

pub  4096R/FE0ADDFA 2023-05-23 [expires: 2025-05-22]
     Key fingerprint = 37D8 BE16 0355 2DA7 BD6A  04D8 C7A0 5F43 FE0A DDFA
uid  AWS SAM CLI Team <aws-sam-cli-signer@amazon.com>
```

La huella digital debe coincidir con la siguiente.

```
37D8 BE16 0355 2DA7 BD6A  04D8 C7A0 5F43 FE0A DDFA
```

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

- Verifique las firmas de la clave pública del firmante:

```
$ gpg --check-sigs FE0ADDFA

pub  4096R/FE0ADDFA 2023-05-23 [expires: 2025-05-22]
uid  AWS SAM CLI Team <aws-sam-cli-signer@amazon.com>
sig!3      FE0ADDFA 2023-05-23  AWS SAM CLI Team <aws-sam-cli-signer@amazon.com>
sig!       73AD885A 2023-05-24  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 comprobar la integridad del instalador del paquete de la CLI de AWS SAM

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. Verifique 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 Tue 30 May 2023 10:03:57 AM UTC using RSA key ID FE0ADDFA
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: 37D8 BE16 0355 2DA7 BD6A 04D8 C7A0 5F43 FE0A DDFA
```

- 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, consulte [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 `aws-sam-cli` GitHub repositorio 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. Ejecute 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: 2023-05-16 20:29:29 +0000
  Certificate Chain:
    1. Developer ID Installer: AMZN Mobile LLC (94KV3E626L)
       Expires: 2027-06-28 22:57:06 +0000
       SHA256 Fingerprint:
           49 68 39 4A BA 83 3B F0 CC 5E 98 3B E7 C1 72 AC 85 97 65 18 B9 4C
           BA 34 62 BF E9 23 76 98 C5 DA
       -----
    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:

```
49 68 39 4A BA 83 3B F0 CC 5E 98 3B E7 C1 72 AC 85 97 65 18 B9 4C BA 34 62 BF E9 23  
76 98 C5 DA
```

Si la cadena de huellas digitales no coincide, no use el instalador de la CLI de AWS SAM. Para llegar al AWS SAM equipo, [crea un problema](#) en el aws-sam-cli GitHub repositorio. Si la cadena de huellas digitales coincide, puede continuar con el instalador del paquete.

Verifique 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 Windows operativo.

Compruebe la integridad del instalador del paquete

1. Haga clic derecho 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.
6. 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.
7. Haga coincidir el valor de la huella digital con el siguiente valor. Si el valor coincide, continúe con la instalación. Si no es así, ponte en contacto con el AWS SAM equipo [creando un problema](#) en el aws-sam-cli GitHub repositorio.

```
c011d416e99a1142c0e0235118ef64c2681f3db9
```

Verifique el valor del hash

Linux

x86_64: instalador de línea de comandos

Compruebe 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 de hash del SHA-256 de 64 caracteres con el de la versión AWS SAMCLI que desee en las notas de la versión](#) AWS SAMCLI en GitHub.

macOS

Instalador de interfaz gráfica de usuario y línea de comandos

Compruebe 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
```

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

Tutorial: Implemente una aplicación Hello World con AWS SAM

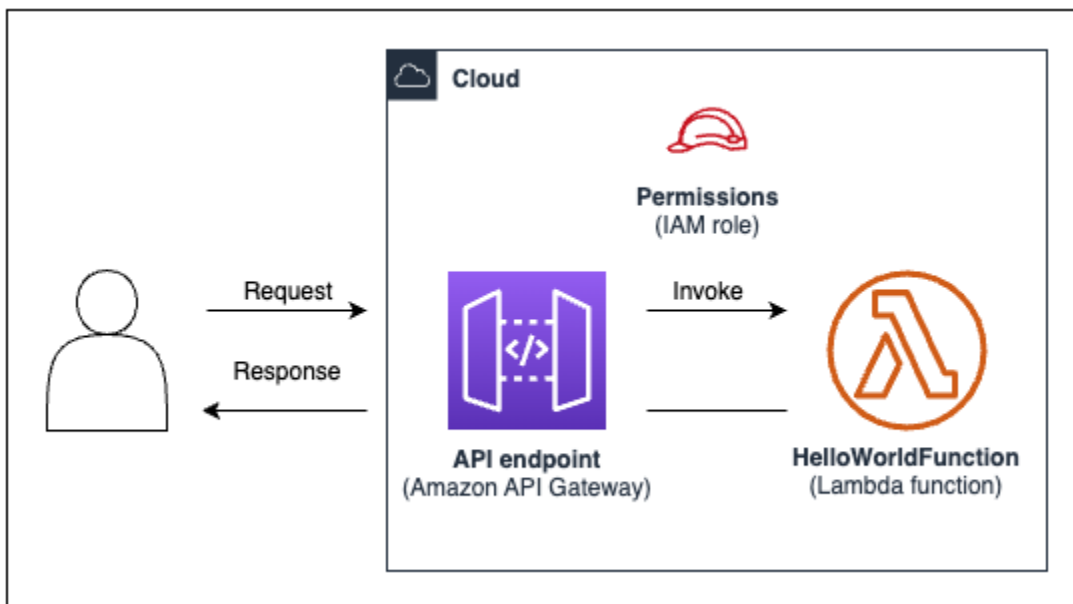
En este tutorial, utilizará la interfaz de línea de AWS Serverless Application Model comandos (AWS SAMCLI) para realizar lo siguiente:

- Inicialice, cree e implemente una aplicación de muestra Hello World
- Realice cambios locales y sincronícelos con AWS CloudFormation.
- Pruebe su aplicación en Nube de AWS.
- Si lo desea, realice pruebas locales en su host de desarrollo.
- Elimine la aplicación de muestra de la Nube de AWS.

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

- Amazon API Gateway: API punto final que utilizará para invocar su función.
- AWS Lambda— Función que procesa la HTTP API GET solicitud y devuelve un hello world mensaje.
- AWS Identity and Access Management (IAM) rol: 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 su aplicación](#)
- [Paso 3: Implemente la aplicación en el 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 su aplicación localmente](#)
- [Paso 8: Elimine la aplicación del Nube de AWS](#)
- [Resolución de problemas](#)
- [Más información](#)

Requisitos previos

Asegúrese de haber completado lo siguiente:

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

Paso 1: inicializar la aplicación de muestra Hello World

En este paso, utilizará la CLI de AWS SAM 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, ejecute lo siguiente desde el directorio de inicio que desee:

```
$ sam init
```

Note

Este comando inicializa la aplicación sin servidor y crea el directorio del proyecto. Este directorio contendrá varios archivos y carpetas. El archivo más importante

`estemplate.yaml`. Esta es tu AWS SAM plantilla. Tu versión de python debe coincidir con la versión de python que aparece en el `template.yaml` archivo que creó el `sam init` comando.

2. La CLI de AWS SAM le guiará a través de la inicialización de una nueva aplicación. Configure lo siguiente:
 1. Seleccione Plantillas de inicio rápido de AWS para elegir una plantilla de inicio.
 2. Seleccione la plantilla Ejemplo de Hello World y descárguela.
 3. Use el tiempo de ejecución de Python y el tipo de paquete zip.
 4. Para este tutorial, opta por no usar el AWS X-Ray rastreo. 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, opte por no configurar el JSON formato de registro estructurado en sus funciones de Lambda.
 7. Asigne a su aplicación el nombre `sam-app`.

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

- Los corchetes (`[]`) indican los valores predeterminados. Deje la respuesta en blanco para seleccionar el valor predeterminado.
- Escriba `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 - 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]: **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 SAMCLI** 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 de la CLI de AWS SAM.

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

```
$ 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 su función de Lambda.
- `hello_world/requirements.txt`: contiene cualquier dependencia de Python que su 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 tiene una aplicación sin servidor creada enteramente en su equipo local!

Paso 2: crear su aplicación

En este paso, use CLI de AWS SAM para crear la aplicación y prepararla para la implementación. Cuando está creando, la CLI de AWS SAM 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 su aplicación

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

```
$ sam build
```

Note

Si no tiene Python en su equipo local, use el comando `sam build --use-container` en su lugar. La CLI de AWS SAM creará un contenedor de Docker que incluye el tiempo de ejecución y las dependencias de su función. Este comando requiere que Docker esté en su equipo local. Para instalar Docker, consulte [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 `.aws-sam` directorio creado por AWS SAM CLI:

```

.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 su función de Lambda. La CLI de AWS SAM crea un directorio para cada función de su aplicación.
- `build/template.yaml`— Contiene una copia de la AWS SAM plantilla a la que se hace referencia AWS 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 CLI de AWS SAM al crear e implementar la aplicación.

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

Paso 3: Implemente la aplicación en el Nube de AWS

Note

Este paso requiere la configuración de AWS las credenciales. Para obtener más información, consulte [Paso 5: utilícelo AWS CLI para configurar AWS las credenciales](#) en [AWS SAM requisitos previos](#).

En este paso, utilice la CLI de AWS SAM para implementar la aplicación en la Nube de AWS. AWS SAMCLI hará lo siguiente:

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

Para implementar la aplicación

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

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

2. Siga el flujo interactivo de la CLI de AWS SAM para configurar los ajustes de la aplicación. Configure lo siguiente:
 1. El nombre de la AWS CloudFormation pila: una pila es un conjunto de AWS recursos que puedes gestionar como una sola unidad. Para obtener más información, consulte [Trabajar con pilas](#) en la Guía del usuario de AWS CloudFormation .
 2. El Región de AWS lugar en el que desplegar tu AWS CloudFormation pila. Para obtener más información, consulte [Puntos de conexión de AWS CloudFormation](#) en la Guía del usuario de AWS CloudFormation .
 3. Para este tutorial, desactive confirmar los cambios antes de la implementación.
 4. Permitir la creación de IAM roles: permite AWS SAM crear el IAM rol necesario para que el recurso de API Gateway y el recurso de función Lambda interactúen.

5. Para este tutorial, desactive deshabilitar la reversión.
6. Permitir HelloWorldFunction sin autorización definida: este mensaje aparece porque el punto final de API Gateway está configurado para ser de acceso público, sin autorización. Dado que esta es la configuración prevista para su aplicación Hello World, deje que la CLI de AWS SAM continúe. Para obtener más información acerca de la autorización de configuración, consulte [Controla el API acceso con tu AWS SAM plantilla](#).
7. Guardar los argumentos en el archivo de configuración: esto actualizará el archivo `samconfig.toml` de la aplicación con sus preferencias de implementación.
8. Seleccione el nombre del archivo de configuración predeterminado.
9. Seleccione el entorno de configuración predeterminado.

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

```

$ 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]: 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 CLI de AWS SAM crea un bucket de Amazon S3 y carga el directorio `.aws-sam`.
- AWS SAM CLI transforma la AWS SAM plantilla en una AWS CloudFormation y la carga en el AWS CloudFormation servicio.
- AWS CloudFormation aprovisiona sus recursos.

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

```
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
```

```
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-
samclisourcebucket-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-15GLOUR9LMT1W

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, enviará una GET solicitud a su API punto final y verá el resultado de la función Lambda.

Para obtener el valor de su API punto final

1. A partir de la información que muestra la CLI de AWS SAM en el paso anterior, localice la sección `Outputs`. En esta sección, localice su `HelloWorldApi` recurso para encontrar el valor de su HTTP punto final. El siguiente es un ejemplo de salida:

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

```
$ 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 su función

- Con su navegador o la línea de comandos, envíe una GET solicitud a su API punto final. 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, utilice la CLI de AWS SAM para invocar la función de Lambda en la Nube de AWS.

Para invocar una función de Lambda en la nube

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

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

3. AWS SAM CLI invoca tu función en la nube y devuelve una respuesta. El siguiente es un ejemplo de salida:

```
$ 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 su línea de comandos, desde el directorio del proyecto `sam-app`, ejecute lo siguiente:

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

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

```
$ 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-
samclisourcebucket-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-15GLOUR9LMT1W

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 su función de Lambda. AWS SAMCLIDetectará automáticamente este cambio y sincronizará la aplicación con el Nube de AWS.

Para modificar y sincronizar su aplicación

1. Abre IDE el `sam-app/hello_world/app.py` archivo que elijas.
2. Cambie el `message` y guarde 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 SAM CLI detecta el cambio y sincroniza la aplicación con el Nube de AWS. El siguiente es un ejemplo de salida:

```
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, vuelva a enviar una GET solicitud a su API dispositivo de punto final.

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

Paso 7: (opcional) probar su aplicación localmente

Note

Este paso es opcional, ya que debe disponer de Docker en su equipo local.

Important

Para usar la CLI de AWS SAM para pruebas locales, debe haber instalado y configurado Docker. Para obtener más información, consulte [Instalar Docker](#).

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

Hará lo siguiente:

1. Cree un entorno local para su función de Lambda e invoquela.
2. Aloje el HTTP API punto final de forma local y utilícelo para invocar la función Lambda.

Para invocar una función de Lambda localmente

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

```
$ sam local invoke
```

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

```
$ 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 su servidor de forma local API

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

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


2. **AWS SAMCLI** crea un Docker contenedor local para la función Lambda y crea un HTTP servidor local para simular el punto final. API El siguiente es un ejemplo de salida:

```
$ 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 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. Con su navegador o la línea de comandos, envíe una GET solicitud a su API punto final 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 su línea de comandos, desde el directorio del proyecto `sam-app`, ejecute lo siguiente:

```
$ sam delete
```

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

```
$ 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 [Solución de problemas para AWS SAM CLI](#).

Más información

Para seguir informándose 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 sobre 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). Estas herramientas se utilizan para:

1. [Desarrolle su aplicación](#) (esto incluye inicializar la aplicación, definir los recursos y crear la aplicación).
2. [Pruebe su aplicación](#).
3. [Depure su aplicación](#).
4. [Implemente su aplicación y sus recursos](#).
5. [Supervise su aplicación](#).

AWS SAM crea el AWS SAM proyecto después de ejecutar el `aws 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 fuentes de eventos y otras propiedades que definen la aplicación sin servidor.

AWS SAMCLI contiene un repositorio de los comandos que utilizas en tu proyecto. AWS SAM 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.

Temas

- [El AWS SAMCLI](#)
- [El AWS SAM proyecto y la AWS SAM plantilla](#)

El AWS SAMCLI

La interfaz de línea de AWS Serverless Application Model comandos (AWS SAMCLI) es la herramienta que se utiliza para ejecutar comandos en el directorio de proyectos de AWS SAM la aplicación y, finalmente, convertirla en una aplicación sin servidor. Más específicamente, AWS

SAMCLI le permite crear, transformar, implementar, depurar, empaquetar, inicializar y sincronizar el directorio de proyectos de su AWS SAM aplicación.

AWS SAM Las plantillas AWS SAMCLI y las plantillas vienen con integraciones de terceros compatibles para crear y ejecutar sus aplicaciones sin servidor.

Temas

- [Cómo se documentan los comandos CLI de AWS SAM.](#)
- [Configuración de la AWS SAM de CLI](#)
- [AWS SAMCLI comandos principales](#)

Cómo se documentan los comandos CLI de AWS SAM.

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 de *texto reemplazable*. 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`** debe proporcionarse tal cual.
3. `template.yaml` se puede reemplazar con su nombre de archivo específico.
4. La salida comienza en `Configuring SAM deploy`.
5. En la salida, `ENTER` e `y` indican los valores reemplazables que usted proporciona.

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. Los ejemplos de la siguiente tabla muestran cómo se pueden optimizar los comandos:

Original	Optimizado 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>

Original	Optimizado con samconfig
<code>sam local start-api --env-vars locals.json --warm-containers EAGER</code>	<code>sam local start-api</code>

AWS SAMCLI 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](#)

Los temas de esta sección le muestran cómo crear [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 \(el samconfig archivo\)](#)
- [Configurar los ajustes del proyecto](#)
- [Configure las credenciales y los ajustes básicos](#)

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

El archivo AWS SAMCLI de configuración (nombre de archivos `samconfig`) 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 `sam deploy -l-guided` comando.

Una vez completada la implementación, el `samconfig` archivo 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 usarlos con. AWS SAMCLI Para obtener más información acerca de este archivo de configuración, consulte [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, consulte [Fundamentos de los archivos de configuración](#).

Para configurar un nuevo entorno

1. Especifique 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. Especifique 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. Utilice 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. Especifique 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, utilice 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, consulte [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
```

Configure las credenciales y los ajustes básicos

Use AWS Command Line Interface (AWS CLI) para configurar los ajustes básicos, como AWS las credenciales, el nombre de la región predeterminado y el formato de salida predeterminado. Una vez configurados, puede usar estos ajustes con la CLI de AWS SAM. Para obtener más información, consulte 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 el AWS CLI](#)
- [Uso de un perfil con nombre habilitado para IAM Identity Center](#)

Para obtener instrucciones de Quick Setup, consulte [Paso 5: utilícelo AWS CLI para configurar AWS las credenciales](#).

AWS SAM CLI comandos principales

AWS SAM CLI 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 AWS SAM CLI comandos, consulte [Referencia de los comandos AWS SAM CLI](#).

Comando	¿Qué hace?	Temas relacionados de
sam build	Prepara una aplicación para los siguientes pasos 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 • sam build
sam deploy	Implementa una aplicación en la AWS nube mediante AWS CloudFormation.	<ul style="list-style-type: none"> • Introducción a la implementación con AWS SAM • sam deploy
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 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 • sam local invoke • sam local start-api • sam local start-lambda
sam remote invoke	Proporciona una forma de acceder a los eventos de prueba que se pueden compartir para las funciones	<ul style="list-style-type: none"> • Introducción a las pruebas en la nube con sam remote invoke • sam remote invoke

Comando	¿Qué hace?	Temas relacionados de
	de AWS Lambda y gestionar los.	
sam remote test-event	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 sam remote test-event • sam remote test-event
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 de la sincronización con Nube de AWS • sam sync

El AWS SAM proyecto y la 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 fuentes de eventos y otras propiedades que definen la aplicación sin servidor.

Note

Un elemento clave de la AWS SAM plantilla es la especificación de la AWS SAM plantilla. Esta especificación proporciona una sintaxis abreviada que, en comparación con ella AWS 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 AWS CloudFormation las plantillas, con tipos de sintaxis únicos que utilizan una sintaxis abreviada con menos líneas de código que. AWS CloudFormation Esto acelera el desarrollo al crear una aplicación sin servidor. Para obtener más información, consulte [AWS SAM recursos y propiedades](#). Para obtener la referencia completa de AWS CloudFormation las plantillas, consulte la [Referencia AWS CloudFormation de plantillas](#) en la Guía del AWS CloudFormation usuario.

Temas

- [AWS SAM anatomía de la plantilla](#)
- [AWS SAM recursos y propiedades](#)
- [AWS CloudFormation Recursos generados para AWS SAM](#)
- [Atributos de recursos compatibles con AWS SAM](#)
- [APIExtensiones de puerta de enlace para AWS SAM](#)
- [Funciones intrínsecas para AWS SAM](#)

AWS SAM anatomía de la plantilla

Un archivo AWS SAM de plantilla sigue de cerca el formato de un archivo de AWS 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 AWS 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 AWS CloudFormation de plantilla como un archivo AWS SAM de plantilla. Para obtener más información, consulte 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 y API sin servidor. Todos los recursos `AWS::Serverless::Function`, `AWS::Serverless::Api` y `AWS::Serverless::SimpleTable` heredan las propiedades que se definen en la sección `Globals`. Para obtener más información, consulte [Sección Globals de la plantilla de AWS SAM](#).
- Sección de recursos. En AWS SAM las plantillas, la `Resources` sección puede contener una combinación de AWS CloudFormation recursos y AWS SAM recursos. Para obtener más información sobre AWS 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 AWS 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.

Puede incluir las secciones de la plantilla en cualquier orden. Sin embargo, si usa extensiones de idioma, debe agregarlas `AWS::LanguageExtensions` antes de la transformación sin servidor (es decir, antes `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 resultar útil utilizar el orden lógico que se muestra en 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, debes incluir esta sección con un valor de `AWS::Serverless-2016-10-31`.

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

[Globals \(opcional\)](#)

Propiedades que son comunes a todas las funciones, tablas simples y API sin servidor. Todos los recursos `AWS::Serverless::Function`, `AWS::Serverless::Api` y `AWS::Serverless::SimpleTable` heredan las propiedades que se definen en la sección `Globals`.

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

[Descripción \(opcional\)](#)

Una cadena de texto que describe la plantilla.

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

[Metadatos \(opcional\)](#)

Objetos que proporcionan información adicional acerca de la plantilla.

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

Parámetros (opcional)

Valores que se van a pasar a su plantilla en tiempo de ejecución (al crear o actualizar una pila). Puede 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`, consulte [sam deploy](#) en la referencia de comandos de la CLI de AWS SAM. Para obtener más información sobre el archivo de configuración, consulte [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 AWS 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 AWS CloudFormation plantillas.

Recursos (obligatorio)

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

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

Siguientes pasos

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](#).

Sección `Globals` de la plantilla de AWS SAM

A veces, los recursos que se declaran en una plantilla de AWS SAM 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 sección `Globals` admite los siguientes tipos de recursos de AWS SAM:

- `AWS::Serverless::Api`
- `AWS::Serverless::Function`
- `AWS::Serverless::HttpApi`
- `AWS::Serverless::SimpleTable`
- `AWS::Serverless::StateMachine`

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

Function:

Architectures:
AssumeRolePolicyDocument:
AutoPublishAlias:
CodeUri:
DeadLetterQueue:
DeploymentPreference:
Description:
Environment:
EphemeralStorage:
EventInvokeConfig:
Handler:
KmsKeyArn:
Layers:
MemorySize:
PermissionsBoundary:
PropagateTags:
ProvisionedConcurrencyConfig:
ReservedConcurrentExecutions:
Runtime:
Tags:
Timeout:
Tracing:
VpcConfig:

HttpApi:

AccessLogSettings:
Auth:
PropagateTags:
StageVariables:
Tags:

SimpleTable:

SSESpecification:

```
StateMachine:  
  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 al declarar una API en la sección de Events. 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 antepone 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 AWS CloudFormation de recursos y propiedades. Para obtener información de referencia sobre todos los tipos de AWS recursos y propiedades AWS 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::Connector](#)

- [AWS::Serverless::Function](#)
- [AWS::Serverless::GraphQLApi](#)
- [AWS::Serverless::HttpApi](#)
- [AWS::Serverless::LayerVersion](#)
- [AWS::Serverless::SimpleTable](#)
- [AWS::Serverless::StateMachine](#)

AWS::Serverless::Api

Creación de una colección de recursos y métodos de Amazon API Gateway que se pueden invocar 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::Function](#) 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 AWS 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 AWS CloudFormation ganchos, consulte [Registrar ganchos](#) en la guía del usuario de la AWS CloudFormation CLI y en el [apigw-enforce-authorizer](#) GitHub repositorio.

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

Note

Al realizar la implementación en AWS CloudFormation, AWS SAM transforma AWS SAM los recursos en AWS CloudFormation recursos. Para obtener más información, consulte [AWS CloudFormation Recursos generados 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
  EndpointConfiguration: EndpointConfiguration
  FailOnWarnings: Boolean
  GatewayResponses: Map
  MergeDefinitions: Boolean
  MethodSettings: MethodSettings
  MinimumCompressionSize: Integer
  Mode: String
  Models: Map
  Name: String
  OpenApiVersion: String
  PropagateTags: Boolean
  StageName: String
  Tags: Map
  TracingEnabled: Boolean
  Variables: Map
```

Propiedades

AccessLogSetting

Configuración de registros de acceso para una etapa.

Tipo: [AccessLogSetting](#)

Obligatorio: no

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

AlwaysDeploy

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

Tipo: Booleano

Obligatorio: no

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

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

Auth

Configure 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 [Controla el API acceso con tu AWS SAM plantilla](#).

Tipo: [ApiAuth](#)

Obligatorio: no

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

BinaryMediaTypes

Lista de tipos MIME que su API puede devolver. Utilice esto para habilitar la compatibilidad con datos binarios para las API. Utilice ~1 en lugar de/en los tipos MIME.

Tipo: lista

Obligatorio: no

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

CacheClusterEnabled

Indica si el almacenamiento 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

AWS 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

AWS 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

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

Cors

Gestione el uso compartido de recursos entre orígenes (CORS) para todas sus API de API Gateway. Especifique el dominio que desea permitir como cadena o especifique 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` el para activar CORS.

Para obtener más información sobre CORS, consulte [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

AWS CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un AWS 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, use la transformación `AWS::Include`. Para obtener más información, consulte [Cargue archivos locales en el momento de la implementación](#).

Tipo: JSON

Obligatorio: no

AWS 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

AWS 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

AWS 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, especifique `True`.

Tipo: Booleano

Obligatorio: no

AWS 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

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

EndpointConfiguration

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

Tipo: [EndpointConfiguration](#)

Obligatorio: no

AWS 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.

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

AWS 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

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

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

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

MinimumCompressionSize

Permita la compresión de los cuerpos de respuesta en función del encabezado Accept-Encoding 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 identity.

Tipo: entero

Obligatorio: no

AWS 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, consulte la propiedad [Modo](#) en el tipo de recurso [AWS::ApiGateway::RestApi](#).

Valores válidos: `overwrite` o `merge`

Tipo: cadena

Obligatorio: no

AWS 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

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

Name

Un nombre para el RestApi recurso API Gateway

Tipo: cadena

Obligatorio: no

AWS 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, consulte 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

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

PropagateTags

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

Tipo: Booleano

Obligatorio: no

Valor predeterminado: `False`

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

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](#), consulte [AWS CloudFormation recursos generados cuando AWS::Serverless::Api se especifica](#). Para obtener información general sobre AWS CloudFormation los recursos generados, consulte [AWS CloudFormation Recursos generados para AWS SAM](#).

Tipo: cadena

Obligatorio: sí

AWS 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, consulte la [etiqueta de recursos](#) en la Guía del usuario de AWS CloudFormation .

Tipo: mapa

Obligatorio: no

AWS CloudFormation compatibilidad: esta propiedad es similar a la [Tags](#) propiedad de un `AWS::ApiGateway::Stage` recurso. La propiedad Tags de SAM consta de pares clave-valor; en CloudFormation ella, consiste en una lista de objetos Tag.

TracingEnabled

Indica si se ha habilitado el rastreo activo con X-Ray para la fase. Para más información acerca de X-Ray, consulte [Rastreo de solicitudes de usuario a API de REST mediante X-Ray](#) en la Guía para desarrolladores de API Gateway.

Tipo: Booleano

Obligatorio: no

AWS 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

AWS 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`, consulte [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`, consulte [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 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, consulte [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}
```

ApiAuth

Configure 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 [Controla el API acceso con tu 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
DefaultAuthorizer: String
InvokeRole: String
ResourcePolicy: ResourcePolicyStatement
UsagePlan: ApiUsagePlan
```

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

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

AWS CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un AWS 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, consulte el tema [Creación y utilización de planes de uso de claves API](#) en la Guía para desarrolladores de API Gateway.

Tipo: Booleano

Obligatorio: no

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

Authorizers

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

Para obtener más información, consulte [Controla el API acceso con tu AWS SAM plantilla](#).

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

Obligatorio: no

Valor predeterminado: Ninguno

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

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

DefaultAuthorizer

Especifique 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

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

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

ResourcePolicy

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

Tipo: [ResourcePolicyStatement](#)

Obligatorio: no

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

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

UsagePlan

Configura un plan de uso asociado a esta API. Para obtener más información sobre claves de API, consulte 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 AWS CloudFormation recursos adicionales cuando se establece: un [AWS::ApiGateway::UsagePlan](#), un [AWS::ApiGateway::UsagePlanKey](#), y un [AWS::ApiGateway::ApiKey](#). Para obtener información acerca de esta situación, consulte [UsagePlanse especifica la propiedad](#). Para obtener información general sobre AWS CloudFormation los recursos generados, consulte [AWS CloudFormation Recursos generados para AWS SAM](#).

Tipo: [ApiUsagePlan](#)

Obligatorio: no

AWS CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un AWS 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"
      }
    }
  }]
IpRangeBlacklist:
  - "10.20.30.40"
```

ApiUsagePlan

Configura un plan de uso para una API de API Gateway. Para obtener más información sobre claves de API, consulte 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 identificadores lógicos `<api-logical-id>UsagePlan`, `<api-logical-id>ApiKey`, y `<api-logical-id>UsagePlanKey`, respectivamente.

SHARED crea [AWS::ApiGateway::UsagePlan](#) y [AWS::ApiGateway::ApiKey](#) recursos que se comparten en cualquier API que también tenga `CreateUsagePlan`: SHARED la misma AWS SAM plantilla. Estos recursos tienen identificadores lógicos `ServerlessUsagePlan`, `ServerlessApiKey`, y `ServerlessUsagePlanKey`, respectivamente. Si usa esta opción, le 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 [Sección Globals de la plantilla de AWS SAM](#).

Valores válidos: PER_API, SHARED y NONE

Tipo: cadena

Obligatorio: sí

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

Description

Es una descripción del plan de uso.

Tipo: cadena

Obligatorio: no

AWS 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

AWS 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

AWS 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

AWS 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

AWS 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, consulte [Controla el API acceso con tu 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
```

Propiedades

AuthorizationScopes

Lista de los ámbitos de autorización de este autorizador.

Tipo: lista

Obligatorio: no

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

Identity

Esta propiedad se puede usar para especificar una solicitud IdentitySource entrante de un autorizador.

Tipo: [CognitoAuthorizationIdentity](#)

Obligatorio: no

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

UserPoolArn

Puede hacer referencia a un grupo de usuarios o especificar un nombre de grupo de usuarios al que desee añadir este autorizador de cognito

Tipo: cadena

Obligatorio: sí

AWS CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un AWS 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 un autorizador IdentitySource en una solicitud entrante. Para obtener más información, IdentitySource consulte la extensión [ApiGateway Authorizer OpenApi](#).

Sintaxis

Para declarar esta entidad en su plantilla de AWS Serverless Application Model (AWS SAM), use 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

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

ReauthorizeEvery

El período time-to-live (TTL), en segundos, que especifica cuánto tiempo API Gateway almacena en caché los resultados del autorizador. Si especifica 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

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

ValidationExpression

Especifique una expresión de validación para la identidad de entrada.

Tipo: cadena

Obligatorio: no

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

Ejemplos

CognitoAuthIdentity

YAML

```
Identity:
  Header: MyCustomAuthHeader
  ValidationExpression: Bearer.*
  ReauthorizeEvery: 30
```

LambdaRequestAuthorizer

Configure una función de Lambda para controlar el acceso a una API con una función de Lambda.

Para obtener más información y ejemplos, consulte [Controla el API acceso con tu 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

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

FunctionArn

Especifique 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

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

FunctionInvokeRole

Añade las credenciales del autorizador a la OpenApi definición del autorizador Lambda.

Tipo: cadena

Obligatorio: no

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

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

AWS CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un AWS 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 un autorizador IdentitySource en una solicitud entrante. Para obtener más información, IdentitySource consulte la extensión [ApiGateway Authorizer OpenApi](#).

Sintaxis

Para declarar esta entidad en su plantilla de AWS Serverless Application Model (AWS SAM), use la siguiente sintaxis:

YAML

```

Context: List
Headers: List
QueryString: 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

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

Headers

Convierte los encabezados en una cadena de expresiones de mapeo de formato `method.request.header.name` separadas por comas.

Tipo: lista

Obligatorio: no

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

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

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

ReauthorizeEvery

El período time-to-live (TTL), en segundos, que especifica cuánto tiempo API Gateway almacena en caché los resultados del autorizador. Si especifica 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

Compatibilidad con AWS CloudFormation: esta propiedad es exclusiva de AWS SAM y no tiene un equivalente AWS 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

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

Ejemplos

LambdaRequestIdentity

YAML

```
Identity:
  QueryStrings:
    - auth
  Headers:
    - Authorization
  StageVariables:
    - VARIABLE
  Context:
    - authcontext
  ReauthorizeEvery: 100
```

LambdaTokenAuthorizer

Configure una función de Lambda para controlar el acceso a una API con una función de Lambda.

Para obtener más información y ejemplos, consulte [Controla el API acceso con tu 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

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

FunctionArn

Especifique 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

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

FunctionInvokeRole

Añade las credenciales del autorizador a la OpenApi definición del autorizador Lambda.

Tipo: cadena

Obligatorio: no

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

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

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

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

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

ValidationExpression

Especifique una expresión de validación para validar la identidad entrante.

Tipo: cadena

Obligatorio: no

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

Ejemplos

LambdaTokenIdentity

YAML

```
Identity:
```

```
Header: MyCustomAuthHeader
ValidationExpression: Bearer.*
ReauthorizeEvery: 30
```

ResourcePolicyStatement

Configure una política de recursos para todos los métodos y rutas de una API. Para obtener más información sobre las políticas de recursos, consulte [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

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

AwsAccountWhitelist

Las AWS cuentas que se van a permitir. Para obtener un ejemplo del uso de esta propiedad, consulte la sección Ejemplos al final de esta página.

Tipo: lista de cadena

Obligatorio: no

AWS CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un AWS 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, consulte la sección Ejemplos al final de esta página.

Tipo: lista

Obligatorio: no

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

IntrinsicVpcBlacklist

La lista de nubes privadas virtuales (VPC) que se van a bloquear, donde cada VPC se especifica como referencia, por ejemplo, una referencia [dinámica](#) o la [función intrínseca Ref](#). Para obtener un ejemplo del uso de esta propiedad, consulte la sección Ejemplos al final de esta página.

Tipo: lista

Obligatorio: no

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

IntrinsicVpcWhitelist

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](#).

Tipo: lista

Obligatorio: no

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

AWS CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un AWS 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, consulte la sección Ejemplos al final de esta página.

Tipo: lista

Obligatorio: no

AWS CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un AWS 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, consulte la sección Ejemplos al final de esta página.

Tipo: lista

Obligatorio: no

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

IpRangeWhitelist

Las direcciones IP o los rangos de direcciones que se van a permitir.

Tipo: lista

Obligatorio: no

AWS CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un AWS 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, consulte la sección Ejemplos al final de esta página.

Tipo: lista

Obligatorio: no

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

AWS CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un AWS 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 su plantilla de AWS Serverless Application Model (AWS SAM), use 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

Compatibilidad con AWS CloudFormation: esta propiedad se transfiere directamente a la propiedad [Bucket](#) del tipo de datos `AWS::ApiGateway::RestApi S3Location`.

Key

La clave de Amazon S3 del archivo de OpenAPI.

Tipo: cadena

Obligatorio: sí

Compatibilidad con AWS CloudFormation: esta propiedad se transfiere directamente a la propiedad [Key](#) del tipo de datos `AWS::ApiGateway::RestApi S3Location`.

Version

Para objetos con control de versiones, la versión del archivo de OpenAPI.

Tipo: cadena

Obligatorio: no

Compatibilidad con AWS CloudFormation: esta propiedad se transfiere directamente a la propiedad [Version](#) del tipo de datos `AWS::ApiGateway::RestApi S3Location`.

Ejemplos

Ejemplo de definición de Uri

Ejemplo de definición de API

YAML

```
DefinitionUri:
```

```
Bucket: mybucket-name  
Key: mykey-name  
Version: 121212
```

CorsConfiguration

Gestione el uso compartido de recursos entre orígenes (CORS) para sus API de API Gateway. Especifique el dominio que desea permitir como cadena o especifique 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` el 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, consulte [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

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

AllowHeaders

Cadena de encabezados que se van a permitir.

Tipo: cadena

Obligatorio: no

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

AllowMethods

Cadena que contiene los métodos HTTP que se van a permitir.

Tipo: cadena

Obligatorio: no

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

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

MaxAge

Cadena que contiene el número de segundos necesarios para almacenar en caché la solicitud CORS Preflight.

Tipo: cadena

Obligatorio: no

AWS CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un AWS 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::Function](#). Si utilizas una integración de proxy Lambda o una integración de proxy HTTP, tu 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: "'www.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': 'www.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 su plantilla de AWS Serverless Application Model (AWS SAM), use la siguiente sintaxis:

YAML

```
BasePath: List
NormalizeBasePath: Boolean
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: /

Compatibilidad de AWS CloudFormation: esta propiedad es similar a la propiedad [BasePath](#) de un recurso `AWS::ApiGateway::BasePathMapping`. AWS SAM crea varios recursos de `AWS::ApiGateway::BasePathMapping`, uno por cada una de las `BasePath` especificadas en esta propiedad.

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.

Use `NormalizeBasePath` con la propiedad de `BasePath`.

Tipo: Booleano

Obligatorio: no

Valor predeterminado: `True`

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

CertificateArn

El nombre de recurso de Amazon (ARN) de un certificado administrado de AWS del punto de conexión de este nombre de dominio. AWS Certificate Manager es el único origen admitido.

Tipo: cadena

Obligatorio: sí

Compatibilidad con AWS CloudFormation: esta propiedad es similar a la propiedad [CertificateArn](#) de un recurso `AWS::ApiGateway::DomainName`. 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`.

Notas adicionales: para un punto de conexión de `EDGE`, debe crear el certificado en la región `us-east-1` de AWS.

DomainName

El nombre de dominio personalizado para su API de API Gateway. No se admiten letras mayúsculas.

AWS SAM genera un recurso [AWS::ApiGateway::DomainName](#) cuando se establece esta propiedad. Para obtener información acerca de esta situación, consulte [DomainName especifica la propiedad](#). Para obtener información sobre los recursos generados de AWS CloudFormation, consulte [AWS CloudFormation Recursos generados para AWS SAM](#).

Tipo: cadena

Obligatorio: sí

Compatibilidad con AWS CloudFormation: esta propiedad se transfiere directamente a la propiedad [DomainName](#) de un recurso `AWS::ApiGateway::DomainName`.

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 propiedad de `CertificateArn` en AWS CloudFormation.

Valores válidos: REGIONAL o EDGE

Tipo: cadena

Obligatorio: no

Valor predeterminado: REGIONAL

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

MutualTlsAuthentication

La configuración de autenticación mutua de seguridad de la capa de transporte (TLS) para un nombre de dominio personalizado.

Tipo: [MutualTlsAuthentication](#)

Obligatorio: no

Compatibilidad con AWS CloudFormation: esta propiedad se transfiere directamente a la propiedad [MutualTlsAuthentication](#) de un recurso `AWS::ApiGateway::DomainName`.

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

Compatibilidad con AWS CloudFormation: esta propiedad se transfiere directamente a la propiedad [OwnershipVerificationCertificateArn](#) de un recurso `AWS::ApiGateway::DomainName`.

Route53

Define una configuración de Amazon Route 53.

Tipo: [Route53Configuration](#)

Obligatorio: no

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

SecurityPolicy

La versión de TLS más conjunto de cifrado para este nombre de dominio.

Tipo: cadena

Obligatorio: no

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

Ejemplos

DomainName

DomainName ejemplo

YAML

```
Domain:
  DomainName: www.example.com
  CertificateArn: arn-example
  EndpointConfiguration: EDGE
  Route53:
    HostedZoneId: Z1PA6795UKMFR9
  BasePath:
    - foo
    - bar
```

Route53Configuration

Configura los conjuntos de registros de Route53 para una API.

Sintaxis

Para declarar esta entidad en su plantilla de AWS Serverless Application Model (AWS SAM), use 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.

Compatibilidad con AWS CloudFormation: esta propiedad se transfiere directamente a la propiedad [DNSName](#) de un recurso `AWS::Route53::RecordSetGroup AliasTarget`.

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

Compatibilidad con AWS CloudFormation: esta propiedad se transfiere directamente a la propiedad [EvaluateTargetHealth](#) de un recurso `AWS::Route53::RecordSetGroup AliasTarget`.

Notas adicionales: No se puede EvaluateTargetHealth establecer como verdadero cuando el alias objetivo es una CloudFront distribución.

HostedZoneId

ID de la zona alojada en la que desea crear registros.

Especifique 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

Compatibilidad con AWS CloudFormation: esta propiedad se transfiere directamente a la propiedad [HostedZoneId](#) de un recurso `AWS::Route53::RecordSetGroup RecordSet`.

HostedZoneName

El nombre de la zona alojada en la que desea crear registros.

Especifique 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

Compatibilidad con AWS CloudFormation: esta propiedad se transfiere directamente a la propiedad [HostedZoneName](#) de un recurso `AWS::Route53::RecordSetGroup` `RecordSet`.

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

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

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 normalmente es un recurso de AWS, como una instancia de EC2 o un equilibrador de carga de ELB, y se hace referencia a él mediante una dirección IP o un nombre de dominio DNS, en función del 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

Compatibilidad con AWS CloudFormation: esta propiedad pasa directamente a la propiedad [Region](#) del tipo de datos RecordSet de AWS::Route53::RecordSetGroup.

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, consulte [Elección de una política de direccionamiento](#) en la Guía para desarrolladores de Amazon Route 53.

Tipo: cadena

Obligatorio: no

Compatibilidad con AWS CloudFormation: esta propiedad pasa directamente a la propiedad [SetIdentifier](#) del tipo de datos RecordSet de AWS::Route53::RecordSetGroup.

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: *ABCDEFGHIJKLMN*

EndpointConfiguration

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

Sintaxis

Para declarar esta entidad en su plantilla de AWS Serverless Application Model (AWS SAM), use la siguiente sintaxis:

YAML

```
Type: String  
VPCEndpointIds: List
```

Propiedades

Type

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

Valores válidos: EDGE, REGIONAL o PRIVATE

Tipo: cadena

Obligatorio: no

Compatibilidad con AWS CloudFormation: esta propiedad se transfiere directamente a la propiedad [Types](#) del tipo de datos `AWS::ApiGateway::RestApi EndpointConfiguration`.

VPCEndpointIds

Una lista de ID de punto de conexión de VPC de una API de REST con la que se crean alias de Route53.

Tipo: lista

Obligatorio: no

Compatibilidad con AWS CloudFormation: esta propiedad se transfiere directamente a la propiedad [VpcEndpointIds](#) del tipo de datos `AWS::ApiGateway::RestApiEndpointConfiguration`.

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 AWS CloudFormation recursos. Para obtener más información, consulte [AWS CloudFormation Recursos generados 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 válida CloudFormation o SAM.

[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

AWS 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

AWS 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

AWS 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

AWS 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 AWS CloudFormation espera a que la pila anidada alcance el `CREATE_COMPLETE` estado. El valor predeterminado es sin tiempo de espera. Cuando AWS 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`, AWS CloudFormation marca la pila anidada como fallida y revierte tanto la pila anidada como la pila principal.

Tipo: entero

Obligatorio: no

AWS 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, consulte [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`, consulte [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/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

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

SemanticVersion

La versión semántica de la aplicación.

Tipo: cadena

Obligatorio: sí

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

Configura permisos entre dos recursos. Para obtener una introducción a los conectores, consulte [Para obtener más información, consulte administración de los permisos de recursos con conectores de AWS SAM](#) ..

Para obtener más información sobre AWS CloudFormation los recursos generados, consulte [AWS CloudFormation recursos generados cuando se especifica AWS::Serverless::Connector](#).

Para enviar comentarios sobre los conectores, [envíe un nuevo número](#) al serverless-application-model AWS GitHub repositorio.

Note

Cuando lo despliegas AWS CloudFormation, AWS SAM transforma tus AWS SAM recursos en AWS CloudFormation recursos. Para obtener más información, consulte [AWS CloudFormation Recursos generados 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::Connector

```
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

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

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

Source

El recurso fuente. Obligatorio cuando se utiliza la sintaxis `AWS::Serverless::Connector`.

Tipo: [ResourceReference](#)

Obligatorio: condicional

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

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

Ejemplos

Conectores integrados

En el siguiente ejemplo, se utilizan conectores integrados para definir una conexión de datos `Write` entre una función de AWS Lambda 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::Connector

En el siguiente ejemplo, se utiliza el [AWS::Serverless::Connector](#) 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::Connector](#) 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::Connector](#) 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::Connector](#).

Note

Para los recursos de la misma plantilla, proporcione la Id. Para recursos que no estén dentro de la misma plantilla, utilice una combinación de otras propiedades. Para obtener más información, consulte [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

AWS CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un AWS 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, proporcione `RoleName` del recurso para que los conectores adjunten las políticas de IAM generadas a un rol de IAM.

Tipo: cadena

Obligatorio: condicional

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

Name

El nombre de un recurso .

Tipo: cadena

Obligatorio: condicional

AWS CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un AWS 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, consulte [API Gateway que invoca una función 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, consulte [AWS SAM referencia de conector](#).

Tipo: cadena

Obligatorio: condicional

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

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

ResourceId

El ID de un recurso. Por ejemplo, el ID de API de la API Gateway.

Tipo: cadena

Obligatorio: condicional

AWS CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un AWS 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, proporcione `RoleName` del recurso para que los conectores adjunten las políticas de IAM generadas a un rol de IAM.

Tipo: cadena

Obligatorio: condicional

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

Type

El AWS CloudFormation tipo de recurso. Para obtener más información, consulte la [Referencia de tipos de recursos y propiedades de AWS](#).

Tipo: cadena

Obligatorio: condicional

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

Ejemplos

API Gateway que invoca una función de Lambda

En el siguiente ejemplo, se utiliza el [AWS::Serverless::Connector](#) recurso para permitir que Amazon API Gateway invoque una AWS Lambda función.

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::Connector](#).

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, consulte [AWS SAM referencia de conector](#).

Tipo: cadena

Obligatorio: condicional

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

Crea una AWS Lambda función, un rol de ejecución AWS Identity and Access Management (IAM) y mapeos de fuentes de eventos que activan la función.

El [AWS::Serverless::Function](#) recurso también admite el atributo `Metadata resource`, por lo que puede indicarle que cree los tiempos de ejecución personalizados AWS SAM que necesite su aplicación. Para obtener más información acerca de la creación de tiempos de ejecución personalizados, consulte [Creación de funciones 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 AWS CloudFormation recursos. Para obtener más información, consulte [AWS CloudFormation Recursos generados 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
  CodeSigningConfigArn: String
  CodeUri: String | FunctionCode
  DeadLetterQueue: Map | DeadLetterQueue
  DeploymentPreference: DeploymentPreference
  Description: String
  Environment: Environment
  EphemeralStorage: EphemeralStorage
  EventInvokeConfig: EventInvokeConfiguration
  Events: EventSource
  FileSystemConfigs: List
  FunctionName: String
  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  
PropagateTags: Boolean  
ProvisionedConcurrencyConfig: ProvisionedConcurrencyConfig  
ReservedConcurrentExecutions: Integer  
Role: String  
RolePath: String  
Runtime: String  
RuntimeManagementConfig: RuntimeManagementConfig  
SnapStart: SnapStart  
Tags: Map  
Timeout: Integer  
Tracing: String  
VersionDescription: String  
VpcConfig: VpcConfig
```

Propiedades

Architectures

La arquitectura del conjunto de instrucciones para la función.

Para obtener más información sobre esta propiedad, consulte [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

AWS 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

AWS 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, consulte [Alias de función de Lambda](#) en la Guía para desarrolladores de AWS Lambda . Para ver ejemplos que utilizan esta propiedad, consulte [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, consulte [AutoPublishAlias se especifica la propiedad](#). Para obtener información general sobre AWS CloudFormation los recursos generados, consulte [AWS CloudFormation Recursos generados para AWS SAM](#).

Tipo: cadena

Obligatorio: no

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

La propiedad requiere que `AutoPublishAlias` se defina.

Si también se especifica `AutoPublishSha256`, su comportamiento tiene prioridad sobre `AutoPublishAliasAllProperties: true`.

Tipo: Booleano

Obligatorio: no

Valor predeterminado: `false`

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

`AutoPublishCodeSha256`

Cuando se usa, esta cadena funciona con el `CodeUri` valor para determinar si es necesario publicar una nueva versión de Lambda. Esta propiedad suele utilizarse para resolver el siguiente problema de despliegue: un paquete de despliegue se almacena en una ubicación de Amazon S3 y se sustituye por un nuevo paquete de despliegue con el código de función Lambda actualizado, pero la `CodeUri` propiedad permanece inalterada (a diferencia de cuando el nuevo paquete de despliegue se carga en una nueva ubicación de Amazon S3 y `CodeUri` se cambia a la nueva ubicación).

Este problema viene marcado por una AWS SAM plantilla que tiene las siguientes características:

- El `DeploymentPreference` objeto está configurado para despliegues graduales (como se describe en [Implementación gradual de aplicaciones sin servidor con AWS SAM](#))
- La `AutoPublishAlias` propiedad está establecida y no cambia entre despliegues
- La `CodeUri` propiedad 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. Especifique 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

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

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, consulte [Configura la firma de código para tu AWS SAM aplicación](#).

Tipo: cadena

Obligatorio: no

AWS CloudFormation compatibilidad: esta propiedad se pasa directamente a la [CodeSigningConfigArn](#) propiedad de un `AWS::Lambda::Function` recurso.

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 proporciona el URI o el objeto [FunctionCode](#) de Amazon S3 de una función, debe hacer referencia a un [paquete de implementación de Lambda](#) válido.

Si proporciona una ruta de archivo local, use la CLI de AWS SAM para cargar el archivo local en el momento de la implementación. Para obtener más información, consulte [Cómo cargar archivos locales durante la implementación con AWS SAMCLI](#).

Si utiliza funciones intrínsecas en la `CodeUri` propiedad, no AWS SAM podrá analizar correctamente los valores. Considere usar la [transformación AWS::Language de extensiones](#) en su lugar.

Tipo: [Cadena | [FunctionCode](#)]

Obligatorio: condicional. Cuando `PackageType` se establece en `Zip`, se requiere `CodeUri` o `InlineCode`.

AWS 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 cola de mensajes fallidos, consulte [Cola de mensajes fallidos](#) en la Guía para desarrolladores de 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

AWS CloudFormation compatibilidad: esta propiedad es similar a la [DeadLetterConfig](#) propiedad de un `AWS::Lambda::Function` recurso. En AWS 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

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

Consulte también: Para obtener más información acerca de esta propiedad, consulte [Implementación gradual de aplicaciones sin servidor con AWS SAM](#).

Description

Una descripción de la función.

Tipo: cadena

Obligatorio: no

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

Environment

La configuración de registro para el entorno de tiempo de ejecución.

Tipo: [entorno](#)

Obligatorio: no

AWS 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, consulte [Entorno de ejecución de Lambda](#) en la Guía para desarrolladores de AWS Lambda .

Tipo: [EphemeralStorage](#)

Obligatorio: no

AWS 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

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

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

FileSystemConfigs

Lista de [FileSystemConfig](#) objetos que especifican la configuración de conexión de un sistema de archivos Amazon Elastic File System (Amazon EFS).

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.

Tipo: lista

Obligatorio: no

AWS 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 especifica un nombre, se genera un nombre único.

Tipo: cadena

Obligatorio: no

AWS CloudFormation compatibilidad: esta propiedad se pasa directamente a la [FunctionName](#) 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, consulte [URL de función](#) en la Guía para desarrolladores de AWS Lambda .

Tipo: [FunctionUrlConfig](#)

Obligatorio: no

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

AWS 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, consulte [Usar imágenes de contenedor con Lambda](#) en la Guía para desarrolladores de AWS Lambda .

Tipo: [ImageConfig](#)

Obligatorio: no

AWS 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, consulte [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

AWS 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

AWS CloudFormation compatibilidad: esta propiedad se pasa directamente a la [ZipFile](#) propiedad del tipo de `AWS::Lambda::Function` Code 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

AWS 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.

Tipo: lista

Obligatorio: no

AWS 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

AWS CloudFormation compatibilidad: esta propiedad se pasa 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

AWS 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, consulte [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, consulte [sam deploy](#).

Valores válidos: `Zip` o `Image`

Tipo: cadena

Obligatorio: no

Valor predeterminado: `Zip`

AWS 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

AWS 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 al rol de ejecución predeterminado 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:

- [Plantillas de política de AWS SAM](#).
- 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

AWS CloudFormation compatibilidad: esta propiedad es similar a la [Policies](#) propiedad de un `AWS::IAM::Role` recurso.

PropagateTags

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

Tipo: Booleano


Obligatorio: no

Valor predeterminado: False

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

AWS CloudFormation compatibilidad: esta propiedad se pasa directamente a la [ProvisionedConcurrencyConfig](#) propiedad de un `AWS::Lambda::Alias` recurso.

ReservedConcurrentExecutions

El número máximo de ejecuciones simultáneas que quiere reservar para la función.

Para obtener más información, consulte [Escalado de funciones de Lambda](#) en la AWS Lambda Guía para desarrolladores de .

Tipo: entero

Obligatorio: no

AWS 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

AWS CloudFormation compatibilidad: esta propiedad es similar a la [Role](#) propiedad de un `AWS::Lambda::Function` recurso. Esto es obligatorio en AWS 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

AWS 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, consulte [Creación de funciones Lambda con tiempos de ejecución personalizados en AWS SAM](#).

Tipo: cadena

Obligatorio: condicional

AWS 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, consulte las [actualizaciones del tiempo de ejecución de Lambda](#) en la Guía para desarrolladores de AWS Lambda .

Tipo: [RuntimeManagementConfig](#)

Obligatorio: no

AWS 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

AWS CloudFormation compatibilidad: esta propiedad se pasa directamente a la [SnapStart](#) propiedad de un `AWS::Lambda::Function` recurso.

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, consulte 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

AWS 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 AWS 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.

Timeout

El tiempo máximo en segundos durante el cual la función se puede poner en marcha.

Tipo: entero

Obligatorio: no

Predeterminado: 3

AWS CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [Timeout](#) propiedad de un `AWS::Lambda::Function` recurso.

Tracing

La cadena que especifica el modo de rastreo de X-Ray de la función.

- `Active`: activa el rastreo de X-Ray para la función.
- `Disabled`: desactiva X-Ray para la función.
- `PassThrough`: activa el rastreo de X-Ray para 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 acerca de X-Ray, consulte [Uso AWS Lambda con AWS X-Ray](#) en la Guía para AWS Lambda desarrolladores.

Valores válidos: `[Active|Disabled|PassThrough]`

Tipo: cadena

Obligatorio: no

AWS 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

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

VpcConfig

La configuración que permite a esta función acceder a los recursos privados dentro de su nube privada virtual (VPC).

Tipo: [VpcConfig](#)

Obligatorio: no

AWS 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`, consulte [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`, consulte [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::Function](#) 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

A continuación se muestra un ejemplo de un [AWS::Serverless::Function](#) de un tipo de paquete Zip (predeterminado) que utiliza InlineCode, Layers, Tracing, Policies, Amazon EFS y un origen de eventos Api.

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:::my-bucket/*'
  Events:
    ApiEvent:
      Type: Api
      Properties:
        Path: /path

```

```
Method: get
```

ImageConfigejemplo

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:4c459dd0104ee29ec65dcad056c0b3ddbe20d6db76b265ade7eda9a066859b1e
    UpdateRuntimeOn: Manual
```

SnapStartejemplos

Ejemplo de una función Lambda SnapStart activada para futuras versiones:

```
TestFunc
Type: AWS::Serverless::Function
Properties:
  ...
  SnapStart:
    ApplyOn: PublishedVersions
```

DeadLetterQueue

Especifica una SQS cola o un SNS tema al que AWS Lambda (Lambda) envía eventos cuando no puede procesarlos. Para obtener más información sobre la funcionalidad de las colas de mensajes fallidos, consulte [Colas de mensajes fallidos](#) en la Guía para desarrolladores de AWS Lambda .

SAMañadirá automáticamente el permiso correspondiente a la función de ejecución de la función de Lambda para permitir que el servicio Lambda acceda al recurso. sqs: se SendMessage añadirá para las colas y SNS:Publish para los temas. SQS SNS

Sintaxis

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

YAML

```
TargetArn: String
Type: String
```

Propiedades

TargetArn

El nombre del recurso de Amazon (ARN) de una SQS cola de Amazon o un SNS tema de Amazon.

Tipo: cadena

Obligatorio: sí

AWS CloudFormation compatibilidad: esta propiedad se transfiere 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

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

Ejemplos

DeadLetterQueue

Ejemplo de Dead Letter Queue para un tema. 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::Function` 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 quieres que se activen por 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

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

Enabled

Si esta preferencia de implementación está habilitada.

Tipo: Booleano

Obligatorio: no

Valor predeterminado: True

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

Hooks

Funciones de Lambda de validación que se ejecutan antes y después del desvío del tráfico.

[Tipo: enlaces](#)

Obligatorio: no

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

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

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

AWS 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

AWS CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un AWS 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 la función intrínseca Fn: :If

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

```

Hooks

Funciones de Lambda de validación que se ejecutan antes y después del desvío del 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 AWS CloudFormation del 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

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

PreTraffic

Función de Lambda que se ejecuta antes del desvío del tráfico.

Tipo: cadena

Obligatorio: no

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

Ejemplos

Enlaces

Ejemplo de funciones de enlace

YAML

```
Hooks:  
  PreTraffic:  
    Ref: PreTrafficLambdaFunction  
  PostTraffic:
```

Ref: PostTrafficLambdaFunction

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

AWS 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

AWS 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

AWS 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

AWS 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

AWS 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

AWS 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, Lambda y EventBridge.

Tipo: cadena

Obligatorio: no

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

Notas adicionales: si el tipo es SQS/SNS y la propiedad de Destination se dejan en blanco, SAM genera automáticamente el recurso SQS/SNS. 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 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

AWS 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`, `Lambda` y `EventBridge`.

Tipo: cadena

Obligatorio: no

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

Notas adicionales: si el tipo es SQS/SNS y la propiedad de Destination se dejan en blanco, SAM genera automáticamente el recurso SQS/SNS. 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 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, consulte 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](#) | | [Cognito](#) | [CloudWatchEvent](#) | [CloudWatchLogs](#) | [DocumentDB](#) | [DynamoDB](#) | | [IoRule](#) | [Kinesis](#) | [MQ](#) | [EventBridgeRule](#)??? | [HttpApi](#)??? | [MSK](#) | [S3](#) | [Programación](#) | [ScheduleV2](#) | [SNS](#) | [SQS](#) | [SelfManagedKafka](#)

Obligatorio: sí

AWS CloudFormation Compatibilidad: esta propiedad es exclusiva de y no tiene una equivalente.
AWS SAM AWS 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

AWS CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un AWS 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, consulte [Configurar el activador de una función de Lambda](#) en la documentación del Alexa Skills Kit.

Tipo: cadena

Obligatorio: no

AWS CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un AWS 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 `Status Code` 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  
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

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

Method

Método HTTP para el que se invoca esta función.

Tipo: cadena

Obligatorio: sí

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

Path

Ruta de URI para la que se invoca esta función. Debe empezar con `/`.

Tipo: cadena

Obligatorio: sí

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

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

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

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

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)

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

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

AuthorizationScopes

Los ámbitos de autorización que se aplicarán a esta API, ruta y método.

Los ámbitos que especifique anularán los ámbitos aplicados por la propiedad de `DefaultAuthorizer` si los ha especificado.

Tipo: lista

Obligatorio: no

AWS CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un AWS 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, consulte [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, consulte [OverrideApiAuth](#).

Valores válidos: `AWS_IAMNONE`, o el identificador lógico de cualquier autorizador definido en la AWS SAM plantilla.

Tipo: cadena

Obligatorio: no

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

InvokeRole

Especifica qué `InvokeRole` utilizar para la autorización de `AWS_IAM`.

Tipo: cadena

Obligatorio: no

Valor predeterminado: `CALLER_CREDENTIALS`

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

Especifique 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, consulte [Anula un autorizador global cuando se especifique for `DefinitionBody` `AWS::Serverless::Api`](#).

Tipo: Booleano

Obligatorio: no

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

ResourcePolicy

Configura la política de recursos para esta ruta en una API.

Tipo: [ResourcePolicyStatement](#)

Obligatorio: no

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

Ejemplos

Autenticación de funciones

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

Puede 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 tu AWS Lambda función, puedes especificar `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 especifique for `DefinitionBody` `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

Configure una política de recursos para todos los métodos y rutas de una API. Para obtener más información sobre las políticas de recursos, consulte [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

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

AwsAccountWhitelist

Las AWS cuentas que se van a permitir. Para obtener un ejemplo del uso de esta propiedad, consulte la sección Ejemplos al final de esta página.

Tipo: lista de cadena

Obligatorio: no

AWS CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un AWS 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, consulte la sección Ejemplos al final de esta página.

Tipo: lista

Obligatorio: no

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

IntrinsicVpcBlacklist

La lista de nubes privadas virtuales (VPC) que se van a bloquear, donde cada VPC se especifica como referencia, por ejemplo, una referencia [dinámica](#) o la [función intrínseca Ref](#). Para obtener un ejemplo del uso de esta propiedad, consulte la sección Ejemplos al final de esta página.

Tipo: lista

Obligatorio: no

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

IntrinsicVpcWhitelist

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](#).

Tipo: lista

Obligatorio: no

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

AWS CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un AWS 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, consulte la sección Ejemplos al final de esta página.

Tipo: lista

Obligatorio: no

AWS CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un AWS 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, consulte la sección Ejemplos al final de esta página.

Tipo: lista

Obligatorio: no

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

IpRangeWhitelist

Las direcciones IP o los rangos de direcciones que se van a permitir.

Tipo: lista

Obligatorio: no

AWS CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un AWS 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, consulte la sección Ejemplos al final de esta página.

Tipo: lista

Obligatorio: no

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

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

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

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

ValidateBody

Especifica si API Gateway utiliza el Model para validar el cuerpo de la solicitud. Para obtener más información, consulte [Habilitar la validación de solicitudes en API Gateway](#) en la Guía para desarrolladores de API Gateway.

Tipo: Booleano

Obligatorio: no

AWS CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un AWS 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, consulte [Habilitar la validación de solicitudes en API Gateway](#) en la Guía para desarrolladores de API Gateway.

Tipo: Booleano

Obligatorio: no

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

Ejemplos

Modelo de solicitud

Ejemplo de modelo de solicitud

YAML

```
RequestModel:
```

```
Model: User
Required: true
ValidateBody: true
ValidateParameters: true
```

RequestParameter

Configure 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

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

Required

Este campo especifica si un parámetro es necesario.

Tipo: Booleano

Obligatorio: condicional

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

AWS 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

AWS 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

AWS 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

AWS 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

AWS CloudFormation compatibilidad: esta propiedad se pasa 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

AWS 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 [Sintaxis de patrones y filtros](#).

Tipo: cadena

Obligatorio: sí

AWS 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

AWS 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

AWS 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

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

AWS 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

AWS 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

AWS 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

AWS 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

AWS 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, consulte [Filtrado de eventos de Lambda](#) en la Guía para desarrolladores de AWS Lambda .

Tipo: [FilterCriteria](#)

Obligatorio: no

AWS 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

AWS CloudFormation compatibilidad: esta propiedad se pasa directamente a la [FullDocument](#) propiedad de un tipo de `AWS::Lambda::EventSourceMapping` `DocumentDBEventSourceConfig` datos.

MaximumBatchingWindowInSeconds

La cantidad de tiempo máxima para recopilar registros antes de invocar la función, en segundos.

Tipo: Entero

Obligatorio: no

AWS 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

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

AWS 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

AWS 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

AWS 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
MaximumBatchingWindowInSeconds: Integer
MaximumRecordAgeInSeconds: Integer
MaximumRetryAttempts: Integer
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

AWS 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

AWS 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

AWS 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

AWS 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, consulte [Filtrado de eventos de AWS Lambda](#) en la Guía para desarrolladores de AWS Lambda .

Tipo: [FilterCriteria](#)

Obligatorio: no

AWS 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, consulte [Informes de errores de artículos en lotes](#) en la Guía para desarrolladores de AWS Lambda .

Valores válidos: ReportBatchItemFailures

Tipo: lista

Obligatorio: no

AWS CloudFormation compatibilidad: esta propiedad se pasa directamente a la [FunctionResponseTypes](#) 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

AWS 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

AWS 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

AWS CloudFormation compatibilidad: esta propiedad se pasa directamente a la [MaximumRetryAttempts](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

ParallelizationFactor

El número de lotes de cada partición que se procesarán simultáneamente.

Tipo: entero

Obligatorio: no

AWS 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

AWS 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

AWS 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

AWS 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, consulte [Ventanas móviles](#) en la Guía para desarrolladores de AWS Lambda .

Tipo: entero

Obligatorio: no

AWS CloudFormation compatibilidad: esta propiedad se pasa directamente a la [TumblingWindowInSeconds](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

Ejemplos

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.

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 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::Function](#) 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 propiedad `DeadLetterQueue` de la función, consulte [Colas de mensajes fallidos](#) en la Guía para desarrolladores de AWS Lambda .

Tipo: [DeadLetterConfig](#)

Obligatorio: no

AWS 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

AWS 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

AWS 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

AWS 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. Puede 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

AWS CloudFormation compatibilidad: esta propiedad se pasa 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

AWS CloudFormation compatibilidad: esta propiedad se pasa 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

AWS CloudFormation compatibilidad: esta propiedad se pasa directamente a la [RetryPolicy](#) propiedad del tipo de `AWS::Events::Rule` Target datos.

RuleName

El nombre de la regla.

Tipo: cadena

Obligatorio: no

AWS 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

Tipo: cadena

Obligatorio: no

AWS 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. Puede 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

AWS 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
    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, al enviar un evento a una función de Lambda que no existe o cuando EventBridge no tiene 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::Function](#) 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 propiedad `DeadLetterQueue` de la función, consulte [Colas de mensajes fallidos](#) en la AWS Lambda Guía para desarrolladores.

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

AWS 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

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

Especifique la propiedad `Type` o la propiedad `Arn`, pero no ambas.

Valores válidos: `SQS`

Tipo: cadena

Obligatorio: no

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

AWS CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Id](#) propiedad del tipo de `AWS::Events::Rule Target` datos.

Ejemplos

Destino

YAML

```
EBRule:
  Type: EventBridgeRule
  Properties:
    Target:
      Id: MyTarget
```

HttpApi

El objeto que describe una fuente de eventos con su 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

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

AWS CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un AWS 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 de conexión diferente a esta función de Lambda. Solo puede existir una de estas rutas predeterminadas por API.

Tipo: cadena

Obligatorio: no

AWS CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un AWS 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 determinada 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 determinadas por API.

Tipo: cadena

Obligatorio: no

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

AWS CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un AWS 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](#) la Guía para desarrolladores de API Gateway.

Nota: Si `RouteSettings` se especifican tanto en la fuente de `HttpApi` recursos como en la fuente de eventos, las AWS SAM fusiona teniendo prioridad con las propiedades de la fuente de eventos.

Tipo: [RouteSettings](#)

Obligatorio: no

AWS CloudFormation compatibilidad: esta propiedad se pasa 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

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

Ejemplos

HttpApi Evento predeterminado

HttpApi Evento que usa la ruta predeterminada. 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.

Configure 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

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

AWS CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un AWS 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 su plantilla. Para obtener más información, consulte [Sección Globals de la plantilla de AWS SAM](#).

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 regla. AWS IoT Para obtener más información, consulte la [Documentación de AWS 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

AWS CloudFormation compatibilidad: esta propiedad se pasa directamente a la [AwsIotSqlVersion](#) propiedad de un recurso. `AWS::IoT::TopicRule TopicRulePayload`

Sql

El enunciado SQL usado para consultar el tema. Para obtener más información, consulte [SQL Reference \(Referencia de SQL de AWS IoT\)](#) en la Guía para desarrolladores de AWS IoT .

Tipo: cadena

Obligatorio: sí

AWS 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
FunctionResponseTypes: List
MaximumBatchingWindowInSeconds: Integer
MaximumRecordAgeInSeconds: Integer
MaximumRetryAttempts: Integer
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

AWS 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

AWS 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

AWS 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

AWS 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, consulte [Filtrado de eventos de AWS Lambda](#) en la Guía para desarrolladores de AWS Lambda .

Tipo: [FilterCriteria](#)

Obligatorio: no

AWS 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, consulte [Informes de errores de artículos en lotes](#) en la Guía para desarrolladores de AWS Lambda .

Valores válidos: `ReportBatchItemFailures`

Tipo: lista

Obligatorio: no

AWS CloudFormation compatibilidad: esta propiedad se pasa directamente a la [FunctionResponseTypes](#) 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

AWS 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

AWS 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

AWS CloudFormation compatibilidad: esta propiedad se pasa directamente a la [MaximumRetryAttempts](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

ParallelizationFactor

El número de lotes de cada partición que se procesarán simultáneamente.

Tipo: entero

Obligatorio: no

AWS 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

AWS 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

AWS 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

AWS 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, consulte [Ventanas móviles](#) en la Guía para desarrolladores de AWS Lambda .

Tipo: entero

Obligatorio: no

AWS CloudFormation compatibilidad: esta propiedad se pasa directamente a la [TumblingWindowInSeconds](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

Ejemplos

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, consulte [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 Lambda en una red pública, el rol 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, consulte [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  
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

AWS 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

AWS 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. Especifique `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`

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

AWS 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, consulte [Filtrado de eventos de AWS Lambda](#) en la Guía para desarrolladores de AWS Lambda .

Tipo: [FilterCriteria](#)

Obligatorio: no

AWS CloudFormation compatibilidad: esta propiedad se pasa directamente a la [FilterCriteria](#) 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

AWS 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

AWS 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

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

AWS 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.

Sintaxis

Para declarar esta entidad en la AWS SAM plantilla, utilice la siguiente sintaxis.

YAML

```

ConsumerGroupId: String
DestinationConfig: DestinationConfig
FilterCriteria: FilterCriteria
MaximumBatchingWindowInSeconds: Integer
SourceAccessConfigurations: SourceAccessConfigurations
StartingPosition: String
StartingPositionTimestamp: Double
Stream: String
Topics: List

```

Propiedades**ConsumerGroupId**

Una cadena que configura cómo se leerán los eventos de los temas de Kafka.

Tipo: cadena

Obligatorio: no

AWS 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

AWS CloudFormation compatibilidad: esta propiedad se pasa directamente a la [DestinationConfig](#) 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, consulte [Filtrado de eventos de AWS Lambda](#) en la Guía para desarrolladores de AWS Lambda .

Tipo: [FilterCriteria](#)

Obligatorio: no

AWS CloudFormation compatibilidad: esta propiedad se pasa directamente a la [FilterCriteria](#) 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

AWS CloudFormation compatibilidad: esta propiedad se pasa directamente a la [MaximumBatchingWindowInSeconds](#) 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

AWS 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.

Valores válidos: AT_TIMESTAMP | LATEST | TRIM_HORIZON

Tipo: cadena

Obligatorio: no

AWS 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

AWS 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

AWS 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

AWS CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Topics](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

Ejemplos

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
```

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

AWS 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

Es el evento del bucket de Amazon S3 para el que se invoca la función de Lambda. Consulte los [tipos de eventos admitidos con Amazon S3](#) para obtener una lista de valores válidos.

Tipo: Cadenas | Lista

Obligatorio: sí

AWS CloudFormation compatibilidad: esta propiedad se pasa 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, consulte [Configuración de notificaciones de eventos de Amazon S3](#) en la Guía del desarrollador del usuario de Amazon Simple Storage Service.

Tipo: [NotificationFilter](#)

Obligatorio: no

AWS CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Filter](#) propiedad del tipo de `AWS::S3::Bucket LambdaConfiguration` datos.

Ejemplos

Evento de S3

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 recurso [AWS::Events::Rule](#) 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 Scheduler es 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

Te recomendamos que lo utilices EventBridge Scheduler para invocar los objetivos según un cronograma. Para definir este tipo de origen de eventos en sus plantillas de AWS SAM, consulte [ScheduleV2](#).

Sintaxis

Para declarar esta entidad en su plantilla de AWS Serverless Application Model (AWS SAM), use 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 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::Function](#) 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 propiedad `DeadLetterQueue` de la función, consulte [Colas de mensajes fallidos](#) en la Guía para desarrolladores de AWS Lambda.

Tipo: [DeadLetterConfig](#)

Obligatorio: no

Compatibilidad con AWS CloudFormation: esta propiedad es similar a la propiedad [DeadLetterConfig](#) del tipo de datos `AWS::Events::RuleTarget`. La versión AWS

SAM de esta propiedad incluye subpropiedades adicionales, por si quiere que AWS SAM cree automáticamente una cola de mensajes fallidos.

Description

Una descripción del conjunto de reglas.

Tipo: cadena

Obligatorio: no

Compatibilidad con AWS CloudFormation: esta propiedad se transfiere directamente a la propiedad [Description](#) 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

Compatibilidad con AWS CloudFormation: esta propiedad es similar a la propiedad [State](#) de un recurso `AWS::Events::Rule`. Si esta propiedad se establece en `true` entonces AWS SAM pasa; de lo contrario `ENABLED`, pasa `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

Compatibilidad con AWS CloudFormation: esta propiedad se transfiere directamente a la propiedad [Input](#) de un recurso `AWS::Events::Rule Target`.

Name

El nombre de la regla. Si no especifica un nombre, AWS CloudFormation genera un único ID físico y utiliza el ID para el nombre de la regla.

Tipo: cadena

Obligatorio: no

Compatibilidad con AWS CloudFormation: esta propiedad se transfiere directamente a la propiedad [Name](#) de un recurso `AWS::Events::Rule`.

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

Compatibilidad con AWS CloudFormation: esta propiedad se transfiere directamente a la propiedad [RetryPolicy](#) 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, consulte [Expresiones de programación para reglas](#).

Tipo: cadena

Obligatorio: sí

Compatibilidad con AWS CloudFormation: esta propiedad se transfiere directamente a la propiedad [ScheduleExpression](#) de un recurso `AWS::Events::Rule`.

State

El estado de la regla.

Valores aceptados DISABLED | ENABLED:

Note

Especifique la propiedad `Enabled` o `State`, pero no ambas.

Tipo: cadena

Obligatorio: no

Compatibilidad con AWS CloudFormation: esta propiedad se transfiere directamente a la propiedad [State](#) de un recurso `AWS::Events::Rule`.

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, al enviar un evento a una función de Lambda que no existe o cuando EventBridge no tiene 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::Function](#) 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 propiedad `DeadLetterQueue` de la función, consulte [Colas de mensajes fallidos](#) en la AWS Lambda Guía para desarrolladores.

Sintaxis

Para declarar esta entidad en su plantilla de AWS Serverless Application Model (AWS SAM), use 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

Compatibilidad con AWS CloudFormation: esta propiedad se transfiere directamente a la propiedad [Arn](#) del tipo de datos `AWS::Events::RuleDeadLetterConfig`.

QueueLogicalId

Se especifica el nombre personalizado de la cola de mensajes fallidos que AWS SAM crea si se especifica `Type`.

Note

Si no se especifica la propiedad `Type`, entonces esta propiedad se pasa por alto.

Tipo: cadena

Obligatorio: no

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

Type

Tipo de cola. Cuando se establece esta propiedad, AWS SAM crea automáticamente una cola de mensajes fallidos y adjunta la [política basada en los recursos](#) necesaria para conceder permiso al recurso de regla para enviar eventos a la cola.

Note

Especifique la propiedad `Type` o la propiedad `Arn`, pero no ambas.

Valores válidos: SQS

Tipo: cadena

Obligatorio: no

Compatibilidad con AWS CloudFormation: esta propiedad es exclusiva de AWS SAM y no tiene un equivalente AWS 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 recurso [AWS::Scheduler::Schedule](#) cuando se establece este tipo de evento.

Sintaxis

Para declarar esta entidad en su plantilla de AWS Serverless Application Model (AWS SAM), use 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 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::Function](#) 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 propiedad `DeadLetterQueue` de la función, consulte [Colas de mensajes fallidos](#) en la Guía para desarrolladores de AWS Lambda.

Tipo: [DeadLetterConfig](#)

Obligatorio: no

Compatibilidad con AWS CloudFormation: esta propiedad es similar a la propiedad [DeadLetterConfig](#) del tipo de datos `AWS::Scheduler::ScheduleTarget`. La versión AWS SAM de esta propiedad incluye subpropiedades adicionales, por si quiere que AWS SAM cree automáticamente una cola de mensajes fallidos.

Description

Una descripción del programa.

Tipo: cadena

Obligatorio: no

Compatibilidad con AWS CloudFormation: esta propiedad se transfiere directamente a la propiedad [Description](#) 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 especifique o antes de ella.

Tipo: cadena

Obligatorio: no

Compatibilidad con AWS CloudFormation: esta propiedad se transfiere directamente a la propiedad [EndDate](#) de un recurso `AWS::Scheduler::Schedule`.

FlexibleTimeWindow

Permite configurar una ventana en la cual se puede invocar una programación.

Tipo: [FlexibleTimeWindow](#)

Obligatorio: no

Compatibilidad con AWS CloudFormation: esta propiedad se transfiere directamente a la propiedad [FlexibleTimeWindow](#) de un recurso `AWS::Scheduler::Schedule`.

GroupName

El nombre de grupo de programación asociado a esta programación. Si no está definido, se utiliza el grupo predeterminado.

Tipo: cadena

Obligatorio: no

Compatibilidad con AWS CloudFormation: esta propiedad se transfiere directamente a la propiedad [GroupName](#) de un recurso `AWS::Scheduler::Schedule`.

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

Compatibilidad con AWS CloudFormation: esta propiedad se transfiere directamente a la propiedad [Input](#) de un recurso `AWS::Scheduler::Schedule Target`.

KmsKeyArn

El ARN de una clave de KMS que se utiliza para cifrar datos personalizados.

Tipo: cadena

Obligatorio: no

Compatibilidad con AWS CloudFormation: esta propiedad se transfiere directamente a la propiedad [KmsKeyArn](#) de un recurso `AWS::Scheduler::Schedule`.

Name

El nombre de la programación. Si no especifica un nombre, AWS SAM genera uno en el formato *Function-Logical-IDEvent-Source-Name* y utiliza ese identificador para el nombre de la programación.

Tipo: cadena

Obligatorio: no

Compatibilidad con AWS CloudFormation: esta propiedad se transfiere directamente a la propiedad [Name](#) de un recurso `AWS::Scheduler::Schedule`.

OmitName

De forma predeterminada, AWS SAM genera y usa un nombre de programación con el formato *< event-source-name ><Function-logical-ID>*. Establezca esta propiedad `true` para que AWS CloudFormation genere un identificador físico único y utilícelo en su lugar para el nombre del programa.

Tipo: Booleano

Obligatorio: no

Valor predeterminado: `false`

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

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 al rol de IAM de destino del programador.

Tipo: cadena

Obligatorio: no

Compatibilidad con AWS CloudFormation: esta propiedad se transfiere directamente a la propiedad [PermissionsBoundary](#) de un recurso `AWS::IAM::Role`.

RetryPolicy

Objeto de la RetryPolicy que incluye información acerca de la configuración de la política de reintentos.

Tipo: [RetryPolicy](#)

Obligatorio: no

Compatibilidad con AWS CloudFormation: esta propiedad se transfiere directamente a la propiedad [RetryPolicy](#) del tipo de datos `AWS::Scheduler::ScheduleTarget`.

RoleArn

El ARN de la función de IAM que EventBridge Scheduler utilizará para el destino cuando se invoque la programación.

Tipo: [RoleArn](#)

Obligatorio: no

Compatibilidad con AWS CloudFormation: esta propiedad se transfiere directamente a la propiedad [RoleArn](#) del tipo de datos `AWS::Scheduler::ScheduleTarget`.

ScheduleExpression

La expresión de programación que determina cuándo y con qué frecuencia se ejecuta el evento programado.

Tipo: cadena

Obligatorio: sí

Compatibilidad con AWS CloudFormation: esta propiedad se transfiere directamente a la propiedad [ScheduleExpression](#) de un recurso `AWS::Scheduler::Schedule`.

ScheduleExpressionTimezone

La zona horaria en la que se evalúa la expresión de programación.

Tipo: cadena

Obligatorio: no

Compatibilidad con AWS CloudFormation: esta propiedad se transfiere directamente a la propiedad [ScheduleExpressionTimezone](#) de un recurso `AWS::Scheduler::Schedule`.

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 especifique o después de ella.

Tipo: cadena

Obligatorio: no

Compatibilidad con AWS CloudFormation: esta propiedad se transfiere directamente a la propiedad [StartDate](#) de un recurso `AWS::Scheduler::Schedule`.

State

Estado de la programación del programador.

Valores aceptados DISABLED | ENABLED:

Tipo: cadena

Obligatorio: no

Compatibilidad con AWS CloudFormation: esta propiedad se transfiere directamente a la propiedad [State](#) de un recurso `AWS::Scheduler::Schedule`.

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

```

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.

Sintaxis

Para declarar esta entidad en la AWS SAM plantilla, utilice la siguiente sintaxis.

YAML

```

BatchSize: Integer
ConsumerGroupId: String
DestinationConfig: DestinationConfig
Enabled: Boolean
FilterCriteria: FilterCriteria
KafkaBootstrapServers: List
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

AWS CloudFormation compatibilidad: esta propiedad se pasa directamente a la [BatchSize](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

Mínimo: 1

Máximo: 10000

ConsumerGroupId

Cadena que configura cómo se leerán los eventos de los temas de Kafka.

Tipo: cadena

Obligatorio: no

AWS CloudFormation compatibilidad: esta propiedad se pasa directamente a la [SelfManagedKafkaConfiguration](#) 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

AWS 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

AWS 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, consulte [Filtrado de eventos de AWS Lambda](#) en la Guía para desarrolladores de AWS Lambda .

Tipo: [FilterCriteria](#)

Obligatorio: no

AWS 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

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

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: BASIC_AUTH | CLIENT_CERTIFICATE_TLS_AUTH | SASL_SCRAM_256_AUTH | SASL_SCRAM_512_AUTH | SERVER_ROOT_CA_CERTIFICATE

Tipo: lista de [SourceAccessConfiguration](#)

Obligatorio: sí

AWS 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.

Valores válidos: `AT_TIMESTAMP` | `LATEST` | `TRIM_HORIZON`

Tipo: cadena

Obligatorio: no

AWS 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

AWS 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

AWS CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Topics](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

Ejemplos

Origen de eventos Kafka autoadministrado

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
```

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

AWS CloudFormation compatibilidad: esta propiedad se pasa 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`

AWS 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, consulte [Uso de colas de mensajes fallidos en Amazon SQS](#) en la Guía para desarrolladores de Amazon Simple Queue Service.

Tipo: Json

Obligatorio: no

AWS 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

AWS 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

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

Topic

El ARN del tema al que se suscribe.

Tipo: cadena

Obligatorio: sí

AWS 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

Especifique 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

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

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

QueueArn

Especifique un arn de cola de SQS existente.

Tipo: cadena

Obligatorio: sí

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

QueuePolicyLogicalId

Proporcione un nombre LogicalId personalizado para el recurso. [AWS::SQS::QueuePolicy](#)

Tipo: cadena

Obligatorio: no

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

QueueUrl

Especifique la URL de la cola asociada a la propiedad QueueArn.

Tipo: cadena

Obligatorio: sí

AWS CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un AWS 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  
MaximumBatchingWindowInSeconds: Integer  
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

AWS 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

AWS 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, consulte [Filtrado de eventos de AWS Lambda](#) en la Guía para desarrolladores de AWS Lambda .

Tipo: [FilterCriteria](#)

Obligatorio: no

AWS 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, consulte [Informes de errores de artículos en lotes](#) en la Guía para desarrolladores de AWS Lambda .

Valores válidos: `ReportBatchItemFailures`

Tipo: lista

Obligatorio: no

AWS CloudFormation compatibilidad: esta propiedad se pasa directamente a la [FunctionResponseTypes](#) 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

AWS CloudFormation compatibilidad: esta propiedad se pasa directamente a la [MaximumBatchingWindowInSeconds](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

Queue

El ARN de la cola.

Tipo: cadena

Obligatorio: sí

AWS 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

AWS CloudFormation compatibilidad: esta propiedad se pasa directamente a la [ScalingConfig](#) propiedad de un `AWS::Lambda::EventSourceMapping` recurso.

Ejemplos

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"]}'
```

Configure 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 Lambda con un evento SQS que tiene el escalado configurado

```
MyFunction:
  Type: AWS::Serverless::Function
  Properties:
    ...
  Events:
    MySQLSEvent:
      Type: SQS
      Properties:
        ...
        ScalingConfig:
          MaximumConcurrency: 10
```

FunctionCode

El [paquete de implementación](#) para una función 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
Version: String
```

Propiedades

Bucket

Un bucket de Amazon S3 en la misma AWS región que su función.

Tipo: cadena

Obligatorio: sí

AWS CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [S3Bucket](#) propiedad del tipo de `AWS::Lambda::Function` datos.

Key

La clave de Amazon S3 del paquete de implementación.

Tipo: cadena

Obligatorio: sí

AWS CloudFormation compatibilidad: esta propiedad se pasa directamente a la [S3Key](#) propiedad del tipo de `AWS::Lambda::Function` 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

AWS CloudFormation compatibilidad: esta propiedad se pasa directamente a la [S3ObjectVersion](#) propiedad del tipo de `AWS::Lambda::Function` datos.

Ejemplos

FunctionCode

CodeUri: Ejemplo de código de función

YAML

```
CodeUri:
  Bucket: mybucket-name
  Key: mykey-name
  Version: 121212
```

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 función que cree utiliza la versión \$LATEST de la función de 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 tu plantilla AWS Serverless Application Model (AWS SAM), usa 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

AWS 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

AWS 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

AWS::Serverless::GraphQLApi

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

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

Auth

Configure la autenticación para tu GraphQL API.

Tipo: [Auth](#)

Obligatorio: sí

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

Cache

La entrada de una operación CreateApiCache.

Tipo: [AWS::AppSync::ApiCache](#)

Obligatorio: no

AWS CloudFormation compatibilidad: esta propiedad se pasa directamente al [AWS::AppSync::ApiCache](#) recurso.

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

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

DomainName

Es el nombre de dominio personalizado para la API GraphQL.

Tipo: [AWS::AppSync::DomainName](#)

Obligatorio: no

AWS CloudFormation compatibilidad: esta propiedad se pasa directamente al [AWS::AppSync::DomainName](#) recurso. AWS SAM genera automáticamente el [AWS::AppSync::DomainNameApiAssociation](#) recurso.

Functions

Configure funciones en las API de GraphQL para realizar ciertas operaciones.

Tipo : [Función](#)

Obligatorio: sí

AWS CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un AWS 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, especifique lo siguiente:

```
Logging: false
```

Tipo: [LogConfig](#)

Obligatorio: no

AWS 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

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

Name

El nombre de la API GraphQL. Especifique esta propiedad para anular el valor `LogicalId`.

Tipo: cadena

Obligatorio: no

AWS 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

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

SchemaInline

La representación de texto de un esquema de GraphQL en formato SDL.

Tipo: cadena

Obligatorio: condicional. Debe especificar `SchemaInline` o `SchemaUri`.

AWS 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, AWS CloudFormation requiere que el archivo se cargue primero en Amazon S3 antes de la implementación. Puede utilizar la CLI de AWS SAM para facilitar este proceso. Para obtener más información, consulte [Cómo cargar archivos locales durante la implementación con AWS SAMCLI](#).

Tipo: cadena

Obligatorio: condicional. Debe especificar `SchemaInline` o `SchemaUri`.

AWS 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

AWS CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Tag](#) propiedad de un `AWS::AppSync::GraphQLApi` recurso.

XrayEnabled

Indique si desea utilizar el [rastreo de AWS de rayos X](#) para este recurso.

Tipo: Booleano

Obligatorio: no

AWS CloudFormation compatibilidad: esta propiedad se pasa directamente a la [XrayEnabled](#) propiedad de un `AWS::AppSync::GraphQLApi` recurso.

Ejemplos

GraphQL API con origen 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

```
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),
  };
}
```

preprocessPostItem.js

```
import { util } from "@aws-appsync/utils";

export function request(ctx) {
  const id = util.autoId();
  const { ...values } = ctx.args;
  values.ups = 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;
}
```

API de GraphQL con una función de Lambda como origen de datos

En este ejemplo, creamos una API de GraphQL que usa una función de Lambda como origen 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. Especifique si desea anular el valor de `LogicalId`.

Tipo: cadena

Obligatorio: sí

AWS 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

AWS 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

AWS 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

AWS CloudFormation compatibilidad: esta propiedad se pasa directamente a la [ApiKeyId](#) propiedad de un `AWS::AppSync::ApiKey` recurso.

Auth

Configure 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

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

LambdaAuthorizer

Especifique la configuración de autorización opcional para su autorizador de funciones de Lambda. Puede configurar esta propiedad opcional si Type se especifica como AWS_LAMBDA.

Tipo: [LambdaAuthorizerConfig](#)

Obligatorio: no

AWS CloudFormation compatibilidad: esta propiedad se pasa directamente a la [LambdaAuthorizerConfig](#) propiedad de un `AWS::AppSync::GraphQLApi` recurso.

OpenIDConnect

Especifique la configuración de autorización opcional para su servicio OpenID Connect compatible. Puede configurar esta propiedad opcional si Type se especifica como OPENID_CONNECT.

Tipo: [OpenID ConnectConfig](#)

Obligatorio: no

AWS 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, consulte [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

AWS CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [AuthenticationType](#) propiedad de un recurso. `AWS::AppSync::GraphQLApi`

UserPool

Especifique la configuración de autorización opcional para utilizar grupos de usuarios de Amazon Cognito. Puede configurar esta propiedad opcional si `Type` se especifica como `AMAZON_COGNITO_USER_POOLS`.

Tipo: [UserPoolConfig](#)

Obligatorio: no

AWS CloudFormation compatibilidad: esta propiedad se pasa directamente a la [UserPoolConfig](#) propiedad de un `AWS::AppSync::GraphQLApi` recurso.

Ejemplos

Configure 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 su plantilla AWS Serverless Application Model (AWS SAM), utilice la siguiente sintaxis.

YAML

```
LambdaAuthorizer: LambdaAuthorizerConfig
OpenIDConnect: OpenIDConnectConfig
Type: String
UserPool: UserPoolConfig
```

Propiedades

LambdaAuthorizer

Especifique la configuración de autorización opcional para su autorizador de AWS Lambda funciones. Puede configurar esta propiedad opcional si Type se especifica como AWS_LAMBDA.

Tipo: [LambdaAuthorizerConfig](#)

Obligatorio: no

AWS CloudFormation compatibilidad: esta propiedad se pasa directamente a la [LambdaAuthorizerConfig](#) propiedad de un `AWS::AppSync::GraphQLApi` [AdditionalAuthenticationProvider](#) objeto.

OpenIDConnect

Especifique la configuración de autorización opcional para su servicio OpenID Connect compatible. Puede configurar esta propiedad opcional si Type se especifica como OPENID_CONNECT.

Tipo: [OpenID ConnectConfig](#)

Obligatorio: no

AWS 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, consulte [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

AWS CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [AuthenticationType](#) propiedad de un objeto. `AWS::AppSync::GraphQLApi` [AdditionalAuthenticationProvider](#)

UserPool

Especifique la configuración de autorización opcional para utilizar grupos de usuarios de Amazon Cognito. Puede configurar esta propiedad opcional si Type se especifica como AMAZON_COGNITO_USER_POOLS.

Tipo: [UserPoolConfig](#)

Obligatorio: no

AWS CloudFormation compatibilidad: esta propiedad se pasa directamente a la [UserPoolConfig](#) propiedad de un `AWS::AppSync::GraphQLApiAdditionalAuthenticationProvider` objeto.

DataSource

Configure 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, consulte [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

Configure una tabla de DynamoDB como origen de datos para su solucionador de API GraphQL.

Tipo: [DynamoDb](#)

Obligatorio: no

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

Lambda

Se utiliza para configurar una función de Lambda como origen de datos para su solucionador de API GraphQL.

Type: [Lambda](#)

Obligatorio: no

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

DynamoDb

Configure 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

AWS 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

AWS 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

AWS 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. Especifique esta propiedad para anular el valor `LogicalId`.

Tipo: cadena

Obligatorio: no

AWS 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](#). Puede 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, consulte [AWS CloudFormation recursos generados cuando se especifica `AWS::Serverless::Connector`](#).

 Note

Puede especificar `Permissions` o `ServiceRoleArn`, pero no ambos. Si no se especifica ninguno de los dos, 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

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

AWS 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.

Puede especificar `Permissions` o `ServiceRoleArn`, pero no ambos.

Tipo: cadena

Requerido: No. Si no se especifica, AWS SAM aplica el valor predeterminado para `Permissions`.

AWS 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.

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

TableName

El nombre de la tabla.

Tipo: cadena

Obligatorio: sí

AWS CloudFormation compatibilidad: esta propiedad se pasa directamente a la [TableName](#) propiedad de un `AWS::AppSync::DataSource` `DynamoDBConfig` objeto.

UseCallerCredentials

Establézcalo en `true` para utilizar IAM con este origen de datos.

Tipo: Booleano

Obligatorio: no

AWS CloudFormation compatibilidad: esta propiedad se pasa directamente a la [UseCallerCredentials](#) propiedad de un `AWS::AppSync::DataSource` `DynamoDBConfig` objeto.

Versioned

Establézcalo en `true` para utilizar [Detección de conflictos, resolución de conflictos y sincronización](#) con este origen de datos.

Tipo: Booleano

Obligatorio: no

AWS CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Versioned](#) propiedad de un `AWS::AppSync::DataSource DynamoDBConfig` objeto.

Lambda

Configura una AWS Lambda función 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

AWS 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

AWS 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

AWS 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. Especifique esta propiedad para anular el valor `LogicalId`.

Tipo: cadena

Obligatorio: no

AWS 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](#) .

AWS CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [ServiceRoleArn](#) propiedad de un `AWS::AppSync::DataSource` recurso.

Función

Configure 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, AWS CloudFormation requiere que el archivo se cargue primero en Amazon S3 antes de la implementación. Puede utilizar la CLI de AWS SAM para facilitar este proceso. Para obtener más información, consulte [Cómo cargar archivos locales durante la implementación con AWS SAMCLI](#).

Tipo: cadena

Obligatorio: no

AWS 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

AWS 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

AWS 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

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

InlineCode

El código de la función que contiene las funciones de solicitud y respuesta.

Tipo: cadena

Obligatorio: no

AWS 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

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

MaxBatchSize

El número máximo de entradas de solicitud de resolución que se enviarán a una sola función de AWS Lambda en una operación de `BatchInvoke`.

Tipo: entero

Obligatorio: no

AWS CloudFormation compatibilidad: esta propiedad se pasa directamente a la [MaxBatchSize](#) propiedad de un `AWS::AppSync::FunctionConfiguration` recurso.

Name

El nombre de la función. Especifique si desea anular el valor de `LogicalId`.

Tipo: cadena

Obligatorio: no

AWS 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

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

AWS 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

AWS 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

AWS CloudFormation compatibilidad: esta propiedad se pasa directamente a la [RuntimeVersion](#) propiedad de un `AWS::AppSync::FunctionConfiguration` `AppSyncRuntime` objeto.

Solucionador

Configure los resolutores para los campos de su 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

AWS 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, AWS CloudFormation requiere que el archivo se cargue primero en Amazon S3 antes de la implementación. Puede utilizar la CLI de AWS SAM para facilitar este proceso. Para obtener más información, consulte [Cómo cargar archivos locales durante la implementación con AWS SAMCLI](#).

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

AWS CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [CodeS3Location](#) propiedad de un `AWS::AppSync::Resolver` recurso.

FieldName

El nombre de su solucionador. Especifique esta propiedad para anular el valor `LogicalId`.

Tipo: cadena

Obligatorio: no

AWS 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

AWS 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. Use el mismo nombre de campo para `LogicalId`.

Tipo: cadena

Obligatorio: sí

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

MaxBatchSize

El número máximo de entradas de solicitud de resolución que se enviarán a una sola función de AWS Lambda en una operación de BatchInvoke.

Tipo: entero

Obligatorio: no

AWS CloudFormation compatibilidad: esta propiedad se pasa 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

AWS 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. Especifique las funciones mediante un identificador lógico en una lista.

Tipo: lista

Obligatorio: sí

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

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

AWS CloudFormation compatibilidad: esta propiedad se pasa directamente a la [SyncConfig](#) propiedad de un `AWS::AppSync::Resolver` recurso.

Ejemplos

Utilice el código de la función de resolución AWS SAM generado y guarde 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

AWS 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

AWS 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, consulte [Trabajar con API HTTP](#) en la Guía del desarrollador de API Gateway.

Te recomendamos que utilices AWS 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 AWS CloudFormation ganchos, consulte [Registrar ganchos](#) en la guía del usuario de la AWS CloudFormation CLI y en el [apigw-enforce-authorizer](#) GitHub repositorio.

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

Note

Al realizar la implementación en AWS CloudFormation, AWS SAM transforma AWS SAM los recursos en AWS CloudFormation recursos. Para obtener más información, consulte [AWS CloudFormation Recursos generados 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

AWS 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, consulte [Control del acceso a las API HTTP con autorizadores de JWT](#) en la Guía para desarrolladores de API Gateway.

Tipo: [HttpApiAuth](#)

Obligatorio: no

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

CorsConfiguration

Administre el uso compartido de recursos entre orígenes (CORS) para todas las API HTTP Gateway. Especifique el dominio que desea permitir como cadena o especifique 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, consulte [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

AWS CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un AWS 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 anule para determinadas rutas.

Tipo: [RouteSettings](#)

Obligatorio: no

AWS 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

AWS 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 AWS 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 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 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

AWS 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

AWS CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un AWS 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 `DefinitionBody` propiedad en lugar de la `DefinitionUri` propiedad o definirla `x-amazon-apigateway-endpoint-configuration` `disableExecuteApiEndpoint` en su definición de OpenAPI.

Tipo: Booleano

Obligatorio: no

AWS 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

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

AWS 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

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

PropagateTags

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

Tipo: Booleano

Obligatorio: no

Valor predeterminado: False

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

RouteSettings

La configuración de ruta, por ruta, para esta API HTTP. Para obtener más información, consulte [Trabajar con rutas para API HTTP](#) en la Guía para desarrolladores de API Gateway.

Tipo: [RouteSettings](#)

Obligatorio: no

AWS 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`

AWS 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

AWS 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

AWS CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un AWS 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, consulte [Ref](#) en la Guía del usuario de AWS CloudFormation .

Ejemplos

Sencillo HttpApi

El siguiente ejemplo muestra el mínimo necesario para configurar un punto de conexión de la API HTTP respaldado por una función 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 faltan 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 cómo configurar el acceso a las API HTTP, consulte [Controlar y administrar el 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: [OAuth2Authorizer](#) | [LambdaAuthorizer](#)

Obligatorio: no

Valor predeterminado: Ninguno

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

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, especifique un autorizador que haya definido en `Authorizers`.

Tipo: cadena

Obligatorio: no

Valor predeterminado: Ninguno

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

EnableIamAuthorizer

Especifique si desea utilizar la autorización de IAM para la ruta de la API.

Tipo: Booleano

Obligatorio: no

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

Ejemplos

Autorizador de OAuth 2.0

Ejemplo de autorizadores de OAuth 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 función. AWS Lambda

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

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

EnableFunctionDefaultPermissions

De forma predeterminada, el recurso de la API HTTP no tiene permiso para invocar el autorizador Lambda. Especifique 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`

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

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

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

`FunctionInvokeRole`

El ARN del rol de IAM que tiene las credenciales necesarias para API Gateway para invocar a la función autorizadora. Especifique 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, consulte [Creación de un autorizador Lambda](#) en la Guía para desarrolladores de API Gateway.

Tipo: cadena

Obligatorio: no

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

AWS CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un AWS 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, consulte [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

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

Headers

Convierte los encabezados en una lista de expresiones de mapeo en el formato `$request.header.name`.

Tipo: Lista

Obligatorio: no

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

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

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

StageVariables

Convierte las variables de etapa dadas en una lista de expresiones de mapeo en el formato `$stageVariables.stageVariable`.

Tipo: Lista

Obligatorio: no

AWS CloudFormation compatibilidad: esta propiedad es exclusiva AWS SAM y no tiene un AWS 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 OAuth 2.0, también conocido como un JSON Web Token (JWT).

Para obtener más información, consulte [Control del acceso a las API HTTP con autorizadores de 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

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

IdentitySource

Expresión de origen de identidad para este autorizador.

Tipo: cadena

Obligatorio: no

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

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

Ejemplos

Autorizador de OAuth 2.0

Ejemplo de autorizadores de OAuth 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

Administra el intercambio de recursos de origen cruzado (CORS) para las API HTTP. Especifique el dominio que desea permitir como cadena o especifique 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, consulte [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

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

AllowHeaders

Representa una colección de encabezados permitidos.

Tipo: lista

Obligatorio: no

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

AllowMethods

Representa una colección de métodos HTTP permitidos.

Tipo: lista

Obligatorio: no

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

AllowOrigins

Representa una colección de orígenes permitidos.

Tipo: lista

Obligatorio: no

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

ExposeHeaders

Representa una colección de encabezados expuestos.

Tipo: lista

Obligatorio: no

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

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

AWS 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

AWS 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

AWS 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: mybucket-name
  Key: mykey-name
  Version: 121212
```

HttpApiDomainConfiguration

Configura un dominio personalizado para una API.

Sintaxis

Para declarar esta entidad en su plantilla de AWS Serverless Application Model (AWS SAM), use 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: /

Compatibilidad con AWS CloudFormation: esta propiedad es similar a la propiedad [ApiMappingKey](#) de un recurso `AWS::ApiGatewayV2::ApiMapping`. AWS SAM crea varios recursos `AWS::ApiGatewayV2::ApiMapping`, uno por cada valor especificado en esta propiedad.

CertificateArn

El nombre de recurso de Amazon (ARN) de un certificado AWS administrado para el punto de conexión de este nombre de dominio. AWS Certificate Manager es el único origen admitido.

Tipo: cadena

Obligatorio: sí

Compatibilidad con AWS CloudFormation: esta propiedad se transfiere directamente a la propiedad [CertificateArn](#) de un recurso `AWS::ApiGateway2::DomainNameDomainNameConfiguration`.

DomainName

El nombre de dominio personalizado para su API de API Gateway. No se admiten letras mayúsculas.

AWS SAM genera un recurso `AWS::ApiGatewayV2::DomainName` cuando se establece esta propiedad. Para obtener información acerca de esta situación, consulte [DomainName especifica la propiedad](#). Para obtener información sobre los recursos generados de AWS CloudFormation, consulte [AWS CloudFormation Recursos generados para AWS SAM](#).

Tipo: cadena

Obligatorio: sí

Compatibilidad con AWS CloudFormation: esta propiedad se transfiere directamente a la propiedad [DomainName](#) de un recurso `AWS::ApiGateway2::DomainName`.

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 propiedad de `CertificateArn` en AWS CloudFormation.

El único valor válido para las API HTTP es `REGIONAL`.

Tipo: cadena

Obligatorio: no

Valor predeterminado: `REGIONAL`

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

MutualTlsAuthentication

La configuración de autenticación mutua de seguridad de la capa de transporte (TLS) para un nombre de dominio personalizado.

Tipo: [MutualTlsAuthentication](#)

Obligatorio: no

Compatibilidad con AWS CloudFormation: esta propiedad se transfiere directamente a la propiedad [MutualTlsAuthentication](#) de un recurso `AWS::ApiGatewayV2::DomainName`.

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

Compatibilidad con AWS CloudFormation: esta propiedad se transfiere directamente a la propiedad [OwnershipVerificationCertificateArn](#) del tipo de datos `AWS::ApiGatewayV2::DomainNameConfiguration`.

Route53

Define una configuración de Amazon Route 53.

Tipo: [Route53Configuration](#)

Obligatorio: no

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

SecurityPolicy

La versión TLS de la política de seguridad para este nombre de dominio.

El único valor válido para las API HTTP es TLS_1_2.

Tipo: cadena

Obligatorio: no

Compatibilidad con AWS CloudFormation: esta propiedad se transfiere directamente a la propiedad [SecurityPolicy](#) del tipo de datos `AWS::ApiGatewayV2::DomainNameDomainNameConfiguration`.

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 su plantilla de AWS Serverless Application Model (AWS SAM), use 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.

Compatibilidad con AWS CloudFormation: esta propiedad se transfiere directamente a la propiedad [DNSName](#) de un recurso `AWS::Route53::RecordSetGroup AliasTarget`.

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

Compatibilidad con AWS CloudFormation: esta propiedad se transfiere directamente a la propiedad [EvaluateTargetHealth](#) de un recurso `AWS::Route53::RecordSetGroup` `AliasTarget`.

Notas adicionales: No se puede `EvaluateTargetHealth` establecer como verdadero cuando el alias objetivo es una CloudFront distribución.

HostedZoneId

ID de la zona alojada en la que desea crear registros.

Especifique `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

Compatibilidad con AWS CloudFormation: esta propiedad se transfiere directamente a la propiedad [HostedZoneId](#) de un recurso `AWS::Route53::RecordSetGroup` `RecordSet`.

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`.

Especifique `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

Compatibilidad con AWS CloudFormation: esta propiedad se transfiere directamente a la propiedad [HostedZoneName](#) de un recurso `AWS::Route53::RecordSetGroup` `RecordSet`.

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

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

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 normalmente es un recurso de AWS, como una instancia de EC2 o un equilibrador de carga de ELB, y se hace referencia a él mediante una dirección IP o un nombre de dominio DNS, en función del 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

Compatibilidad con AWS CloudFormation: esta propiedad pasa directamente a la propiedad [Region](#) del tipo de datos `RecordSet` de `AWS::Route53::RecordSetGroup`.

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, consulte [Elección de una política de direccionamiento](#) en la Guía para desarrolladores de Amazon Route 53.

Tipo: cadena

Obligatorio: no

Compatibilidad con AWS CloudFormation: esta propiedad pasa directamente a la propiedad [SetIdentifier](#) del tipo de datos `RecordSet` de `AWS::Route53::RecordSetGroup`.

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::LayerVersion

Crea una Lambda `LayerVersion` que contiene el código de biblioteca o de tiempo de ejecución que necesita una función 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, consulte [Creación de capas Lambda en AWS SAM](#).

Nota importante: Desde la publicación del atributo de [UpdateReplacePolicy](#) recurso en AWS 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 AWS CloudFormation recursos. Para obtener más información, consulte [AWS CloudFormation Recursos generados 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
  RetentionPolicy: String
```

Propiedades

CompatibleArchitectures

Especifica las arquitecturas del conjunto de instrucciones compatibles para la versión de capa.

Para obtener más información sobre esta propiedad, consulte [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`

AWS 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

AWS 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 [sam build](#) 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

AWS 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

AWS 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

AWS CloudFormation compatibilidad: esta propiedad es similar a la [LayerName](#) propiedad de un `AWS::Lambda::LayerVersion` recurso. Si no especifica un nombre, se utilizará como nombre el ID lógico del recurso.

LicenseInfo

Información sobre la licencia correspondiente `LayerVersion`.

Tipo: cadena

Obligatorio: no

AWS CloudFormation compatibilidad: esta propiedad se transfiere directamente a la [LicenseInfo](#) propiedad de un `AWS::Lambda::LayerVersion` recurso.

RetentionPolicy

Esta propiedad especifica si las versiones antiguas `LayerVersion` se conservan o se eliminan al eliminar un recurso. Si necesita conservar sus versiones antiguas `LayerVersion` al actualizar o reemplazar un recurso, debe tener el `UpdateReplacePolicy` atributo activado. Para obtener información sobre cómo hacerlo, consulte el [UpdateReplacePolicyatributo](#) de la Guía del AWS CloudFormation usuario.

Valores válidos: `Retain` o `Delete`

Tipo: cadena

Obligatorio: no

AWS CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un AWS 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`, consulte [Ref](#) en la Guía del usuario de AWS CloudFormation .

Ejemplos

`LayerVersionExample`

Ejemplo de `LayerVersion`

YAML

```
Properties:
  LayerName: MyLayer
  Description: Layer description
  ContentUri: 's3://my-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
Version: String
```

Propiedades

Bucket

El bucket de Amazon S3 del archivo de la capa.

Tipo: cadena

Obligatorio: sí

AWS CloudFormation compatibilidad: esta propiedad se pasa directamente a la [S3Bucket](#) propiedad del tipo de `AWS::Lambda::LayerVersionContent` datos.

Key

La clave de Amazon S3 del archivo de la capa.

Tipo: cadena

Obligatorio: sí

AWS CloudFormation compatibilidad: esta propiedad se pasa directamente a la [S3Key](#) propiedad del tipo de `AWS::Lambda::LayerVersionContent` datos.

Version

Para objetos con control de versiones, el objeto de archivo de la capa que se va a utilizar.

Tipo: cadena

Obligatorio: no

AWS CloudFormation compatibilidad: esta propiedad se pasa directamente a la [S3ObjectVersion](#) propiedad del tipo de `AWS::Lambda::LayerVersionContent` datos.

Ejemplos

LayerContent

Ejemplo de contenido de capa

YAML

```
LayerContent:
  Bucket: mybucket-name
  Key: mykey-name
  Version: 121212
```

AWS::Serverless::SimpleTable

Creará 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 utilizar la funcionalidad más avanzada de DynamoDB, utilice un recurso de [AWS::DynamoDB::Table](#) en su lugar.

Note

Cuando despliega en AWS CloudFormation, AWS SAM transforma sus AWS SAM recursos en AWS CloudFormation recursos. Para obtener más información, consulte [AWS CloudFormation Recursos generados 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: ProvisionedThroughput
  SSESpecification: SSESpecification
```

TableName: *String*

Tags: *Map*

Propiedades

PointInTimeRecoverySpecification

La configuración utilizada para habilitar la recuperación a un momento dado.

Tipo: [PointInTimeRecoverySpecification](#)

Obligatorio: no

AWS 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

AWS CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un AWS 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: [ProvisionedThroughput](#)

Obligatorio: no

AWS 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.

Type: [SSESpecification](#)

Obligatorio: no

AWS 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

AWS 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, consulte la [etiqueta de recursos](#) en la Guía del usuario de AWS CloudFormation .

Tipo: mapa

Obligatorio: no

AWS 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, consulte [Ref](#) en la Guía del usuario de AWS CloudFormation .

Ejemplos

SimpleTableExample

Ejemplo de un SimpleTable

YAML

```
Properties:
  TableName: my-table
  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

AWS 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

AWS 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
```

AWS::Serverless::StateMachine

Creación de una máquina de estados de AWS Step Functions que se puede utilizar para organizar AWS Lambda funciones y otros AWS recursos con el fin de crear flujos de trabajo complejos y sólidos.

Para obtener más información acerca de Step Functions, consulte la [AWS Step Functions Guía para desarrolladores de](#) .

Note

Cuando lo despliega AWS CloudFormation, AWS SAM transforma sus AWS SAM recursos en AWS CloudFormation recursos. Para obtener más información, consulte [AWS CloudFormation Recursos generados 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
  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, consulte [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

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

Definition

La definición de la máquina de estados es un objeto, cuyo formato coincide con el formato del archivo de AWS SAM plantilla, por ejemplo, JSON o YAML. Las definiciones de máquina de estado cumplen con el [Lenguaje de estados de Amazon](#).

Para ver un ejemplo de una definición de máquina de estados en línea, consulte [Ejemplos](#).

Debe proporcionar un `Definition` o un `DefinitionUri`.

Tipo: mapa

Obligatorio: condicional

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

DefinitionSubstitutions

Un string-to-string mapa que especifica las asignaciones de las variables de marcador de posición en la definición de la máquina de estados. 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

AWS CloudFormation compatibilidad: esta propiedad es similar a la [DefinitionSubstitutions](#) propiedad de un recurso.

`AWS::StepFunctions::StateMachine` Si se especifica alguna función intrínseca en una definición de máquina de estados en línea, AWS SAM agrega entradas a esta propiedad para insertarlas en 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 proporciona 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

AWS 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, consulte [Gestión de implementaciones continuas con versiones y alias](#) en la Guía para desarrolladores de AWS Step Functions .

Especifique esta propiedad `AutoPublishAlias` antes de configurar. La configuración `DeploymentPreference` se aplicará al alias especificado con `AutoPublishAlias`.

Al especificar `DeploymentPreference`, AWS SAM genera el valor de la `StateMachineVersionArn` subpropiedad automáticamente.

Tipo: [DeploymentPreference](#)

Obligatorio: no

AWS 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

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

Logging

Define qué eventos del historial de ejecución se registran y dónde se registran.

Tipo: [LoggingConfiguration](#)

Obligatorio: no

AWS 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

AWS 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


AWS 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:

- [Plantillas de política de AWS SAM](#).
- 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

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

PropagateTags

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

Tipo: Booleano

Obligatorio: no

Valor predeterminado: `False`

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

AWS 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

AWS CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Path](#) propiedad de un `AWS::IAM::Role` recurso.

Tags

Un string-to-string mapa que especifica las etiquetas añadidas a la máquina de estados y la función de ejecución correspondiente. Para obtener información sobre las claves y los valores válidos de las etiquetas, consulte la propiedad [Etiquetas](#) de un recurso [AWS::StepFunctions::StateMachine](#).

Tipo: mapa

Obligatorio: no

AWS 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 habilitado o no para la máquina de estados. Para obtener más información sobre X-Ray con Step Functions, consulte [Step Functions y AWS X-Ray](#) en la Guía para desarrolladores de AWS Step Functions .

Tipo: [TracingConfiguration](#)

Obligatorio: no

AWS CloudFormation compatibilidad: esta propiedad se pasa 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

AWS 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, consulte [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`, consulte [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

El siguiente es un ejemplo de una definición de máquina de estados en línea que permite que una función lambda invoque una máquina de estados. Tenga en cuenta que en este ejemplo se espera que la `Role` propiedad 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, consulte 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: [Programación](#) | [ScheduleV2](#) | [CloudWatchEvent](#) | [EventBridgeRule](#) | [Api](#)

Obligatorio: sí

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

Type

Tipo de evento.

Valores válidos: `Api`, `Schedule`, `ScheduleV2`, `CloudWatchEvent`, `EventBridgeRule`

Tipo: cadena

Obligatorio: sí

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

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

Method

El método HTTP para el que se invoca esta función.

Tipo: cadena

Obligatorio: sí

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

Path

La ruta URI para la que se invoca esta función. El valor debe empezar por `/`.

Tipo: cadena

Obligatorio: sí

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

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

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

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

AuthorizationScopes

Los ámbitos de autorización que se aplicarán a esta API, ruta y método.

Los ámbitos que especifique anularán los ámbitos aplicados por la propiedad de `DefaultAuthorizer` si los ha especificado.

Tipo: lista

Obligatorio: no

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

Authorizer

El `Authorizer` para una máquina de estados específica.

Si especificó un autorizador global para la API y desea que esta máquina de estados sea pública, anule el autorizador global configurando `Authorizer` en `NONE`.

Tipo: cadena

Obligatorio: no

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

ResourcePolicy

Configura la política de recursos para esta API y esta ruta.

Tipo: [ResourcePolicyStatement](#)

Obligatorio: no

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

Ejemplos

StateMachine-Authenticación

El siguiente ejemplo especifica la autorización a nivel de máquina de estado.

YAML

```
Auth:  
  ApiKeyRequired: true  
  Authorizer: NONE
```

ResourcePolicyStatement

Configure una política de recursos para todos los métodos y rutas de una API. Para obtener más información sobre las políticas de recursos, consulte [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

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

AwsAccountWhitelist

Las AWS cuentas que se van a permitir. Para obtener un ejemplo del uso de esta propiedad, consulte la sección Ejemplos al final de esta página.

Tipo: lista de cadena

Obligatorio: no

AWS CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un AWS 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, consulte la sección Ejemplos al final de esta página.

Tipo: lista

Obligatorio: no

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

IntrinsicVpcBlacklist

La lista de nubes privadas virtuales (VPC) que se van a bloquear, donde cada VPC se especifica como referencia, por ejemplo, una referencia [dinámica](#) o la [función intrínseca Ref](#). Para obtener un ejemplo del uso de esta propiedad, consulte la sección Ejemplos al final de esta página.

Tipo: lista

Obligatorio: no

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

IntrinsicVpcWhitelist

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](#).

Tipo: lista

Obligatorio: no

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

AWS CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un AWS 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, consulte la sección Ejemplos al final de esta página.

Tipo: lista

Obligatorio: no

AWS CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un AWS 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, consulte la sección Ejemplos al final de esta página.

Tipo: lista

Obligatorio: no

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

IpRangeWhitelist

Las direcciones IP o los rangos de direcciones que se van a permitir.

Tipo: lista

Obligatorio: no

AWS CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un AWS 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, consulte la sección Ejemplos al final de esta página.

Tipo: lista

Obligatorio: no

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

AWS CloudFormation compatibilidad: esta propiedad es exclusiva de AWS SAM y no tiene un AWS 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 y CloudWatchEvent usa el mismo servicio, API y AWS 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

AWS 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

AWS 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

AWS 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

AWS CloudFormation compatibilidad: esta propiedad se pasa 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 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

AWS 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

AWS 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

AWS 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

AWS 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. Puede 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

AWS CloudFormation compatibilidad: esta propiedad se pasa 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

AWS CloudFormation compatibilidad: esta propiedad se pasa 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

AWS CloudFormation compatibilidad: esta propiedad se pasa directamente a la [RetryPolicy](#) propiedad del tipo de `AWS::Events::Rule` `Target` datos.

RuleName

El nombre de la regla.

Tipo: cadena

Obligatorio: no

AWS 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

AWS 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. Puede 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

AWS 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

AWS 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

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

Especifique la propiedad Type o la propiedad Arn, pero no ambas.

Valores válidos: SQS

Tipo: cadena

Obligatorio: no

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

AWS CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Id](#) propiedad del tipo de `AWS::Events::Rule Target` datos.

Ejemplos

Destino

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 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

AWS 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

AWS 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

AWS 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

AWS 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, AWS CloudFormation genera un identificador físico único y lo utiliza como nombre de la regla.

Tipo: cadena

Obligatorio: no

AWS 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

AWS CloudFormation compatibilidad: esta propiedad se pasa directamente a la [RetryPolicy](#) propiedad del tipo de `AWS::Events::Rule` Target datos.

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.

AWS CloudFormation compatibilidad: esta propiedad se pasa 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, consulte [Expresiones de programación para reglas](#).

Tipo: cadena

Obligatorio: sí

AWS 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

AWS 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. Puede 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

AWS 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

AWS 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

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

Especifique la propiedad Type o la propiedad Arn, pero no ambas.

Valores válidos: SQS

Tipo: cadena

Obligatorio: no

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

AWS CloudFormation compatibilidad: esta propiedad se pasa directamente a la [Id](#) propiedad del tipo de `AWS::Events::Rule Target` datos.

Ejemplos

Destino

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 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

AWS CloudFormation compatibilidad: esta propiedad es similar a la [DeadLetterConfig](#) propiedad del tipo de `AWS::Scheduler::ScheduleTarget` 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

AWS 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 especifique o antes de ella.

Tipo: cadena

Obligatorio: no

AWS 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

AWS 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

AWS 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

AWS 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

AWS 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

AWS 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 *<State-machine-logical-event-source-name-ID><>*. Establezca esta propiedad `true` para AWS CloudFormation generar un identificador físico único y utilícelo en su lugar para el nombre del programa.

Tipo: Booleano

Obligatorio: no

Valor predeterminado: `false`

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

AWS 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

AWS CloudFormation compatibilidad: esta propiedad se pasa directamente a la [RetryPolicy](#) propiedad del tipo de `AWS::Scheduler::ScheduleTarget` 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

AWS CloudFormation compatibilidad: esta propiedad se pasa directamente a la [RoleArn](#) propiedad del tipo de `AWS::Scheduler::ScheduleTarget` datos.

ScheduleExpression

La expresión de programación que determina cuándo y con qué frecuencia se ejecuta la programación.

Tipo: cadena

Obligatorio: sí

AWS 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

AWS 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 especifique o después de ella.

Tipo: cadena

Obligatorio: no

AWS 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

AWS 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-demo-bucket
    Key: my-state-machine.asl.json
    Version: 3
  Policies:
    - LambdaInvokePolicy:
      FunctionName: !Ref MyFunction
```

AWS CloudFormation Recursos generados para AWS SAM

En esta sección se proporcionan detalles sobre los AWS CloudFormation recursos que se crean al AWS SAM procesar la AWS plantilla. El conjunto de AWS CloudFormation recursos que se AWS SAM genera varía según 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 AWS 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` `AWS::Lambda::Version` recursos AWS SAM adicionales.

En esta sección se enumeran los escenarios y los AWS CloudFormation recursos que generan, y se muestra cómo hacer referencia a AWS CloudFormation los recursos generados en el archivo de AWS SAM plantilla.

Hacer referencia a los recursos de AWS CloudFormation generados.

Tiene dos opciones para hacer referencia a AWS CloudFormation los recursos generados en su archivo de AWS SAM plantilla: por propiedad referenciable `LogicalId` o por propiedad referenciable.

Hacer referencia a los recursos generados mediante AWS CloudFormation `LogicalId`

Cada uno de los AWS CloudFormation recursos que AWS SAM genera tiene un [LogicalId](#) identificador alfanumérico (A-Z, a-z, 0-9) único en un archivo de plantilla. AWS SAM utiliza los AWS SAM recursos `LogicalIds` del archivo de plantilla para construir los recursos que `LogicalIds` genera. AWS CloudFormation Puede usar el `LogicalId` AWS 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 AWS CloudFormation recurso que haya declarado explícitamente. Para obtener más información sobre `LogicalIds` las AWS SAM plantillas AWS 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 AWS Management Console, AWS CLI, o una de las AWS SDKs. No recomendamos referenciar estos recursos con `LogicalId` porque los valores hash pueden cambiar.

Hacer referencia a AWS CloudFormation los recursos generados mediante una 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 AWS CloudFormation recurso generado y a sus propiedades en el archivo de AWS SAM plantilla.

Note

No todos los AWS CloudFormation recursos generados tienen propiedades a las que se puede hacer referencia. Para dichos recursos, debe utilizar los LogicalId.

Escenarios de recursos generados AWS CloudFormation

En la siguiente tabla se resumen los AWS SAM recursos y las propiedades que componen los escenarios que generan AWS CloudFormation recursos. Los temas de la columna Escenarios proporcionan detalles sobre los AWS CloudFormation recursos adicionales que se AWS SAM generan para ese escenario.

AWS SAM recurso	AWS CloudFormation Recurso base	Escenarios
AWS::Serverless::Api	AWS::ApiGateway::RestApi	<ul style="list-style-type: none"> DomainName especifica la propiedad UsagePlan especifica la propiedad
AWS::Serverless::Application	AWS::CloudFormation::Stack	<ul style="list-style-type: none"> Además de generar el AWS CloudFormation recurso base, no hay escenarios adicionales para este recurso sin servidor.
AWS::Serverless::Function	AWS::Lambda::Function	<ul style="list-style-type: none"> AutoPublishAlias se especifica la propiedad Role La propiedad del rol no está especificada DeploymentPreference se especifica la propiedad Events Se ha especificado una fuente de eventos de API HttpApiEvents Se ha especificado una fuente de HttpApi eventos

AWS SAM recurso	AWS CloudFormation Recurso base	Escenarios
		<p><u>Se ha especificado una origen de eventos de transmisión</u></p> <ul style="list-style-type: none"> • <u>Se ha especificado una origen de eventos de puente de eventos (o bus de eventos)</u> • <u>Se ha especificado una fuente de lotRule 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>
<p><u>AWS::Serverless::HttpApi</u></p>	<p><u>AWS::ApiGatewayV2::Api</u></p>	<ul style="list-style-type: none"> • <u>StageName especifica la propiedad</u> • <u>StageName la propiedad no está especificada</u> • <u>DomainName especifica la propiedad</u>
<p><u>AWS::Serverless::LayerVersion</u></p>	<p><u>AWS::Lambda::LayerVersion</u></p>	<ul style="list-style-type: none"> • Además de generar el AWS CloudFormation recurso base, no hay escenarios adicionales para este recurso sin servidor.
<p><u>AWS::Serverless::SimpleTable</u></p>	<p><u>AWS::DynamoDB::Table</u></p>	<ul style="list-style-type: none"> • Además de generar el AWS CloudFormation recurso base, no hay escenarios adicionales para este recurso sin servidor.

AWS SAM recurso	AWS CloudFormation Recurso base	Escenarios
<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 una fuente de eventos de API</u> • <u>Se ha especificado una origen de eventos de puente de eventos (o bus de eventos)</u>

Temas

- [AWS CloudFormation recursos generados cuando AWS::Serverless::Api se especifica](#)
- [AWS CloudFormation recursos generados cuando AWS::Serverless::Application se especifica](#)
- [AWS CloudFormation recursos generados cuando se especifica AWS::Serverless::Connector](#)
- [AWS CloudFormation recursos generados cuando AWS::Serverless::Function se especifica](#)
- [AWS CloudFormation recursos generados cuando AWS::Serverless::GraphQLApi se especifica](#)
- [AWS CloudFormation recursos generados cuando AWS::Serverless::HttpApi se especifica](#)
- [AWS CloudFormation recursos generados cuando AWS::Serverless::LayerVersion se especifica](#)
- [AWS CloudFormation recursos generados cuando AWS::Serverless::SimpleTable se especifica](#)
- [AWS CloudFormation recursos generados cuando AWS::Serverless::StateMachine se especifica](#)

AWS 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 AWS 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 AWS 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 AWS CloudFormation recursos, cuando `AWS::Serverless::Api` se especifica, AWS SAM genera AWS CloudFormation recursos adicionales para los siguientes escenarios.

Escenarios

- [DomainName especifica la propiedad](#)
- [UsagePlan especifica la propiedad](#)

DomainName especifica la propiedad

Cuando se especifica la DomainName Domain propiedad de `AWS::Serverless::Api` una, AWS SAM genera el `AWS::ApiGateway::DomainName` AWS 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 especifica la propiedad

Cuando se especifica la Auth propiedad de `AWS::Serverless::Api` una, AWS SAM genera los siguientes AWS CloudFormation recursos: `AWS::ApiGateway::UsagePlan`, `AWS::ApiGateway::UsagePlanKey`, y `AWS::ApiGateway::ApiKey`.

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`

AWS CloudFormation recursos generados cuando `AWS::Serverless::Application` se especifica

Cuando `AWS::Serverless::Application` se especifica un, AWS Serverless Application Model (AWS SAM) genera un AWS 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 AWS CloudFormation recurso)

AWS 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 recurso `AWS::Serverless::Connector` en una plantilla de AWS SAM, AWS SAM genera los siguientes recursos AWS CloudFormation según sea necesario.

AWS::IAM::ManagedPolicy

LogicalId: `<connector-LogicalId>Policy`

Propiedad referenciable: N/D (para hacer referencia a este recurso AWS CloudFormation, debe usar el `LogicalId`.)

AWS::SNS::TopicPolicy

LogicalId: `<connector-LogicalId>TopicPolicy`

Propiedad referenciable: N/D (para hacer referencia a este recurso AWS CloudFormation, debe usar el `LogicalId`.)

AWS::SQS::QueuePolicy

LogicalId: `<connector-LogicalId>QueuePolicy`

Propiedad referenciable: N/D (para hacer referencia a este recurso AWS CloudFormation, debe usar el `LogicalId`.)

AWS::Lambda::Permission

LogicalId: `<connector-LogicalId><permission>LambdaPermission`

`<permission>` es un permiso especificado por la propiedad `Permissions`. Por ejemplo, `Write`.

Propiedad referenciable: N/D (para hacer referencia a este recurso AWS CloudFormation, debe usar el `LogicalId`.)

AWS 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 AWS 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 AWS CloudFormation recurso)

Además de este AWS CloudFormation recurso, cuando `AWS::Serverless::Function` se especifica, AWS SAM también genera AWS CloudFormation recursos para los siguientes escenarios.

Escenarios

- [AutoPublishAlias se especifica la propiedad](#)
- [La propiedad del rol no está especificada](#)
- [DeploymentPreference se especifica la propiedad](#)
- [Se ha especificado una fuente de eventos de API](#)
- [Se ha especificado una fuente de HttpApi eventos](#)
- [Se ha especificado una origen de eventos de transmisión](#)
- [Se ha especificado una origen de eventos de puente de eventos \(o bus de eventos\)](#)
- [Se ha especificado una fuente de IoTRule eventos](#)
- [OnSuccess\(o OnFailure\) la propiedad está especificada para los eventos de Amazon SNS](#)
- [OnSuccess\(o OnFailure\) la propiedad está especificada para los eventos de Amazon SQS](#)

AutoPublishAlias se especifica la propiedad

Cuando `AWS::Serverless::Function` se especifica la `AutoPublishAlias` propiedad de un, AWS SAM genera los siguientes AWS 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`, `LogicalId` es: *MyFunctionAlias live*.

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, la *MyFunction* versión *926eeb5ff1*.

Propiedad referenciable: *<function-LogicalId>.Version*

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` AWS CloudFormation recurso.

AWS::IAM::Role

LogicalId: <function-LogicalId>Role

Propiedad referenciable: N/A (debe usar la `LogicalId` para hacer referencia a este AWS CloudFormation recurso)

`DeploymentPreference` se especifica la propiedad

Cuando `AWS::Serverless::Function` se especifica la `DeploymentPreference` propiedad de un, AWS SAM genera los siguientes AWS 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` AWS CloudFormation recurso.

AWS::CodeDeploy::Application

LogicalId: ServerlessDeploymentApplication

Propiedad referenciable: N/A (debe usar la `LogicalId` para hacer referencia a este AWS CloudFormation recurso)

AWS::CodeDeploy::DeploymentGroup

LogicalId: *<function-LogicalId>*DeploymentGroup

Propiedad referenciable: N/A (debe usar la para hacer referencia LogicalId a este recurso)
AWS CloudFormation

AWS::IAM::Role

LogicalId: CodeDeployServiceRole

Propiedad referenciable: N/A (debe usar la para hacer referencia LogicalId a este recurso)
AWS CloudFormation

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 AWS CloudFormation recurso.
RestApiId

AWS::ApiGateway::RestApi

LogicalId: ServerlessRestApi

Propiedad referenciable: N/A (debe usar la LogicalId para hacer referencia a este AWS
CloudFormation recurso)

Se ha especificado una fuente de HttpApi eventos

Cuando la Event propiedad de un AWS::Serverless::Function se establece enHttpApi, pero no se especifica, AWS SAM genera el AWS::ApiGatewayV2::Api AWS CloudFormation recurso.
ApiId

AWS::ApiGatewayV2::Api

LogicalId: ServerlessHttpApi

Propiedad referenciable: N/A (debe usar la LogicalId para hacer referencia a este AWS
CloudFormation recurso)

Se ha especificado una origen de eventos de transmisión

Cuando la Event propiedad de an `AWS::Serverless::Function` se establece en uno de los tipos de transmisión, AWS SAM genera el `AWS::Lambda::EventSourceMapping` AWS 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 AWS 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::Function` se establece en uno de los tipos de puente de eventos (o bus de eventos), AWS SAM genera el `AWS::Events::Rule` AWS 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 AWS CloudFormation recurso)

Se ha especificado una fuente de `IoTRule` eventos

Cuando la Event propiedad de an `AWS::Serverless::Function` se establece en `IoTRule`, AWS SAM genera el recurso. `AWS::IoT::TopicRule` AWS CloudFormation

AWS::IoT::TopicRule

LogicalId: <function-LogicalId><event-LogicalId>

Propiedad referenciable: N/A (debe usar la para hacer referencia a este recurso) `LogicalId` AWS CloudFormation

OnSuccess(o OnFailure) la propiedad está especificada para los eventos de Amazon SNS

Cuando se especifica la OnSuccess (oOnFailure) 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 AWS 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)
AWS 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 (oOnFailure) 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 AWS 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)
AWS 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.

AWS CloudFormation recursos generados cuando `AWS::Serverless::GraphQLApi` se especifica

Al especificar un `AWS::Serverless::GraphQLApi` recurso en una plantilla AWS Serverless Application Model (AWS SAM), AWS SAM siempre crea los siguientes recursos base AWS CloudFormation .

AWS::AppSync::DataSource

*LogicalId: <graphqlapi-LogicalId><datasource-RelativeId><datasource-Type>*DataSource

Propiedad referenciable: N/A (debe usar la LogicalId para hacer referencia a este AWS CloudFormation recurso)

AWS::AppSync::FunctionConfiguration

LogicalId: <graphqlapi-LogicalId><function-RelativeId>

Propiedad referenciable: N/A (debe usar la para hacer referencia LogicalId a este recurso) AWS CloudFormation

AWS::AppSync::GraphQLApi

LogicalId: <graphqlapi-LogicalId>

Propiedad referenciable: N/A (debe usar la para hacer referencia LogicalId a este recurso) AWS CloudFormation

AWS::AppSync::GraphQLSchema

*LogicalId: <graphqlapi-LogicalId>*Schema

Propiedad referenciable: N/A (debe usar la para hacer referencia LogicalId a este recurso) AWS CloudFormation

AWS::AppSync::Resolver

LogicalId: <graphqlapi-LogicalId><OperationType><resolver-RelativeId>

Propiedad referenciable: N/A (debe usar la para hacer referencia LogicalId a este recurso)
AWS CloudFormation

Además de estos AWS CloudFormation recursos, cuando se especifique, también AWS::Serverless::GraphQLApi se AWS SAM pueden generar los siguientes AWS CloudFormation recursos.

AWS::AppSync::ApiCache

LogicalId: <graphqlapi-LogicalId>ApiCache

Propiedad referenciable: N/A (debe usar la LogicalId para hacer referencia a este AWS CloudFormation recurso)

AWS::AppSync::ApiKey

LogicalId: <graphqlapi-LogicalId><apikey-RelativeId>

Propiedad referenciable: N/A (debe usar la para hacer referencia LogicalId a este recurso)
AWS CloudFormation

AWS::AppSync::DomainName

LogicalId: <graphqlapi-LogicalId>DomainName

Propiedad referenciable: N/A (debe usar la para hacer referencia LogicalId a este recurso)
AWS CloudFormation

AWS::AppSync::DomainNameApiAssociation

LogicalId: <graphqlapi-LogicalId>DomainNameApiAssociation

Propiedad referenciable: N/A (debe usar la para hacer referencia LogicalId a este recurso)
AWS CloudFormation

AWS SAM también puede usar el AWS::Serverless::Connector recurso para conceder permisos. Para obtener más información, consulte [AWS CloudFormation recursos generados cuando se especifica AWS::Serverless::Connector](#).

AWS CloudFormation recursos generados cuando `AWS::Serverless::HttpApi` se especifica

Cuando `AWS::Serverless::HttpApi` se especifica un, AWS Serverless Application Model (AWS SAM) genera un AWS 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 AWS CloudFormation recurso)

Además de este AWS CloudFormation recurso, cuando `AWS::Serverless::HttpApi` se especifica, AWS SAM también genera AWS 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` AWS 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` en `Gamma`, el `LogicalId` es: `MyHttpApiGammaStage`.

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` AWS CloudFormation recurso.

AWS::ApiGatewayV2::Stage

LogicalId: *<httpapi-LogicalId>*ApiGatewayDefaultStage

Propiedad referenciable: *<httpapi-LogicalId>*.Stage

DomainName especifica la propiedad

Cuando se especifica la DomainName Domain propiedad de AWS::Serverless::HttpApi una, AWS SAM genera el AWS::ApiGatewayV2::DomainName AWS 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

AWS CloudFormation recursos generados cuando AWS::Serverless::LayerVersion se especifica

Cuando AWS::Serverless::LayerVersion se especifica un, AWS Serverless Application Model (AWS SAM) genera un AWS 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 AWS CloudFormation recurso)

AWS CloudFormation recursos generados cuando AWS::Serverless::SimpleTable se especifica

Cuando AWS::Serverless::SimpleTable se especifica un, AWS Serverless Application Model (AWS SAM) genera un AWS 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 AWS CloudFormation recurso)

AWS 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 AWS CloudFormation .

AWS::StepFunctions::StateMachine

LogicalId: <statemachine-LogicalId>

Propiedad referenciable: N/A (debe usar la `LogicalId` para hacer referencia a este AWS CloudFormation recurso)

Además de este AWS CloudFormation recurso, cuando `AWS::Serverless::StateMachine` se especifica, AWS SAM también genera AWS CloudFormation recursos para los siguientes escenarios:

Escenarios

- [La propiedad del rol no está especificada](#)
- [Se ha especificado una fuente 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` AWS CloudFormation recurso.

AWS::IAM::Role

LogicalId: <statemachine-LogicalId>Role

Propiedad referenciable: N/A (debe usar la `LogicalId` para hacer referencia a este AWS CloudFormation recurso)

Se ha especificado una fuente de eventos de API

Cuando la Event propiedad de un `AWS::Serverless::StateMachine` se establece en `Api`, pero no se especifica, AWS SAM genera el `AWS::ApiGateway::RestApi` AWS CloudFormation recurso. `RestApiId`

AWS::ApiGateway::RestApi

LogicalId: `ServerlessRestApi`

Propiedad referenciable: N/A (debe usar la `LogicalId` para hacer referencia a este AWS CloudFormation recurso)

Se ha especificado un origen de eventos de puente de eventos (o bus de eventos)

Cuando la Event propiedad de un `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` AWS 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 AWS CloudFormation recurso)

Atributos de recursos compatibles con AWS SAM

Los atributos de los recursos son atributos que se pueden añadir a los recursos de AWS SAM y AWS CloudFormation para controlar otros comportamientos y relaciones. Para obtener más información sobre los atributos de los recursos, consulte 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 AWS CloudFormation. De los atributos de recursos admitidos, algunos se copian únicamente en el AWS CloudFormation

recurso generado base del AWS SAM recurso correspondiente y otros se copian en todos los AWS CloudFormation recursos generados como resultado del AWS SAM recurso correspondiente. Para obtener más información sobre AWS CloudFormation los recursos generados a partir de AWS SAM los recursos correspondientes, consulte [AWS CloudFormation Recursos generados 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 Metadatos ^{1, 2}	Únicamente el recurso AWS CloudFormation generado por la base. Para obtener información sobre el mapeo entre AWS SAM los recursos y AWS CloudFormation los recursos base, consulte Escenarios de recursos generados AWS CloudFormation .
Condition DeletionPolicy UpdateReplacePolicy	Todos los AWS CloudFormation recursos generados a partir del AWS SAM recurso correspondiente. Para obtener información sobre los escenarios de AWS CloudFormation los recursos generados, consulte Escenarios de recursos generados AWS CloudFormation .

Notas:

1. Para obtener más información sobre el uso del atributo de recurso de Metadata con el tipo de recurso `AWS::Serverless::Function`, consulte [Creación de funciones 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`, consulte [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 AWS 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 AWS 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`

APIExtensiones de puerta de enlace para AWS SAM

Diseñadas específicamente para ello AWS, las extensiones API Gateway proporcionan personalizaciones y funciones adicionales para el diseño y la administración APIs. Se trata de extensiones de la API especificación Open que admiten autorizaciones e API integraciones AWS específicas de Gateway. API

API Las extensiones de Gateway son extensiones de la API especificación Open que admiten la autorización AWS específica y API las integraciones específicas de Gateway. API [Para obtener más información sobre las extensiones de Gateway, consulte API Gateway Extensions to OpenAPI. API](#)

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.

APIExtensión Gateway	Con el apoyo de 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
x-amazon-apigateway-authtype Propiedad	Sí
x-amazon-apigateway-binary-propiedad de -media-types	Sí
x-amazon-apigateway-documentation Objeto	No
x-amazon-apigateway-endpoint-Objeto de configuración	No
x-amazon-apigateway-gatewayObjeto -responses	Sí
x-amazon-apigateway-gateway-respuestas. gatewayResponseObjeto	Sí
x-amazon-apigateway-gateway-respuestas. responseParametersObjeto	Sí
x-amazon-apigateway-gateway-respuestas. responseTemplatesObjeto	Sí
x-amazon-apigateway-integration Objeto	Sí
x-amazon-apigateway-integration. requestTemplates Objeto	Sí
x-amazon-apigateway-integration. requestParametersObjeto	No
x-amazon-apigateway-integrationObjeto .responses	Sí
x-amazon-apigateway-integrationObjeto .response	Sí
x-amazon-apigateway-integration. responseTemplatesObjeto	Sí

x-amazon-apigateway-integration. responseParametersObjeto	Sí
x-amazon-apigateway-request-propiedad del validador	No
x-amazon-apigateway-request-Objeto validators	No
x-amazon-apigateway-request-validadores. requestValidatorObjeto	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 añadir la `AWS::LanguageExtensions` transformación para resolver este problema. `AWS::LanguageExtensions` se trata de una macro alojada por la AWS CloudFormation que se pueden utilizar funciones intrínsecas y otras funcionalidades que, de forma predeterminada, no están incluidas en AWS CloudFormation.

Transform:

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

Note

Nota: Si utiliza funciones intrínsecas en la `CodeUri` propiedad, no AWS SAM podrá analizar correctamente los valores. Considere usar `AWS::LanguageExtensions` transformar en su lugar.

Para obtener más información, consulte la [sección Propiedades de `AWS::Serverless::Function`](#).

Para obtener más información sobre las funciones intrínsecas, consulte la [Referencia de función intrínseca](#) en la Guía del usuario de AWS CloudFormation .

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 en determinados casos de uso, como el trabajo con capas Lambda, el uso de aplicaciones anidadas, el control del acceso a las API de API Gateway, la organización de los recursos con AWS Step Functions y la firma de código de las aplicaciones. AWS SAM A continuación, se enumeran los tres hitos principales que debe completar para desarrollar su aplicación.

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 terminar de leer y [empezarCómo usar AWS Serverless Application Model \(AWS SAM\)](#), estarás listo para crear un AWS SAM proyecto en tu 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 AWS Lambda funciones.

Para crear un AWS SAM proyecto, consulte los temas de estas secciones.

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 CLI de AWS SAM

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 detalla 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 AWS SAM plantilla. Tu versión de python debe coincidir con la versión de python que aparece en el `template.yaml` archivo que creó el `sam init` comando.

Elegir una plantilla inicial

Una plantilla consta de lo siguiente:

1. Una AWS SAM plantilla para el 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 plantilla de inicio AWS 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 local HTTPS o una ruta al 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.](#)

Puede 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 su 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 su 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, consulte [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 SAM CLI Descarga 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
```

Configura el AWS X-Ray rastreo

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, consulte [sam init](#).

Inicialice una aplicación mediante una ubicación de plantilla personalizada

Utilice la opción `--location` y proporcione 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
```

Inicialice una aplicación sin el flujo interactivo

Utilice la opción `--no-interactive` y proporcione 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 [Solución de problemas para AWS SAM CLI](#).

Ejemplos

Inicialice una nueva aplicación sin servidor mediante la plantilla Hello World Starter AWS

Para ver este ejemplo, consulte [Paso 1: inicializar la aplicación de muestra Hello World](#) en el tutorial: Implementación de una aplicación de Hello World.

Inicialice 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
```

El siguiente es 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`, consulte los enlaces siguientes:

- [Aprendizaje AWS SAM: sam init, sobre](#) la serie Serverless Land «Aprendizaje AWS SAM». YouTube
- [Estructuración de aplicaciones sin servidor para usarlas con AWS SAMCLI \(Sesiones con el SAM S2E7\) — Sesiones](#) con la serie siguiente. AWS SAM 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 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 las API de API Gateway, la organización de los recursos con AWS Step Functions, 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](#)
- [Configura y administra el acceso a los recursos en tu AWS SAM plantilla](#)
- [Controla el API acceso con tu 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 EventBridge Scheduler en AWS SAM](#)
- [Organizar los 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, identificas 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 AWS 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
AWS::Serverless::Api	Crea una colección de recursos y métodos de API Gateway que se pueden invocar a través de puntos de enlace HTTPS.
AWS::Serverless::Application	Integrar una aplicación sin servidor desde AWS Serverless Application Repository o desde un bucket de Amazon S3 como una aplicación anidada.
AWS::Serverless::Connector	Configura permisos entre dos recursos. Para obtener una introducción a los conectores, consulte Para obtener más información, consulte administración de los permisos de recursos con conectores de AWS SAM ..
AWS::Serverless::Function	Crea una AWS Lambda función, un rol de ejecución AWS Identity and Access Management (IAM) y mapeos de fuentes de eventos que activan la función.
AWS::Serverless::GraphQLApi	crea y configura una AWS AppSync GraphQL API para su aplicación sin servidor.
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.
AWS::Serverless::LayerVersion	Crea una Lambda LayerVersion que contiene el código de biblioteca o de tiempo de ejecución que necesita una función Lambda.
AWS::Serverless::SimpleTable	Crea una tabla de DynamoDB con una clave principal de atributo único.
AWS::Serverless::StateMachine	Crea una máquina de AWS Step Functions estados que se puede utilizar para organizar AWS Lambda funciones y otros AWS recursos

AWS SAM sintaxis abreviada

Qué hace con un recurso relacionado AWS

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 AWS CloudFormation y el AWS SAM soporte, consulte la [referencia sobre tipos de AWS recursos y propiedades](#) en la Guía del AWS CloudFormation usuario.

Configura y administra el acceso a los recursos en tu 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 AWS SAM de políticas. El último tema proporciona detalles sobre la administración del acceso de los usuarios mediante la misma mecánica que se AWS CloudFormation utiliza para la administración de los usuarios.

Para obtener más información, consulte [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 [Para obtener más información, consulte administración de los permisos de recursos con conectores de AWS SAM ..](#)

Plantillas de política de AWS SAM

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 AWS Lambda funciones, 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 [Plantillas de política de AWS SAM.](#)

AWS CloudFormation mecanismos

AWS 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 AWS 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 Utilice 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, consulte [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, consulte [Plantillas de política de AWS SAM](#).

- Para todos los demás escenarios, o cuando se requiera granularidad, utilice AWS CloudFormation mecanismos.

Para obtener más información, consulte administración de los permisos de recursos con conectores de AWS SAM .

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 los conectores AWS SAM

Al redactar 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 recién se están iniciando en el desarrollo sin servidores o para los desarrolladores experimentados que desean aumentar su velocidad de desarrollo.

Uso de conectores AWS SAM

Utilice 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
```

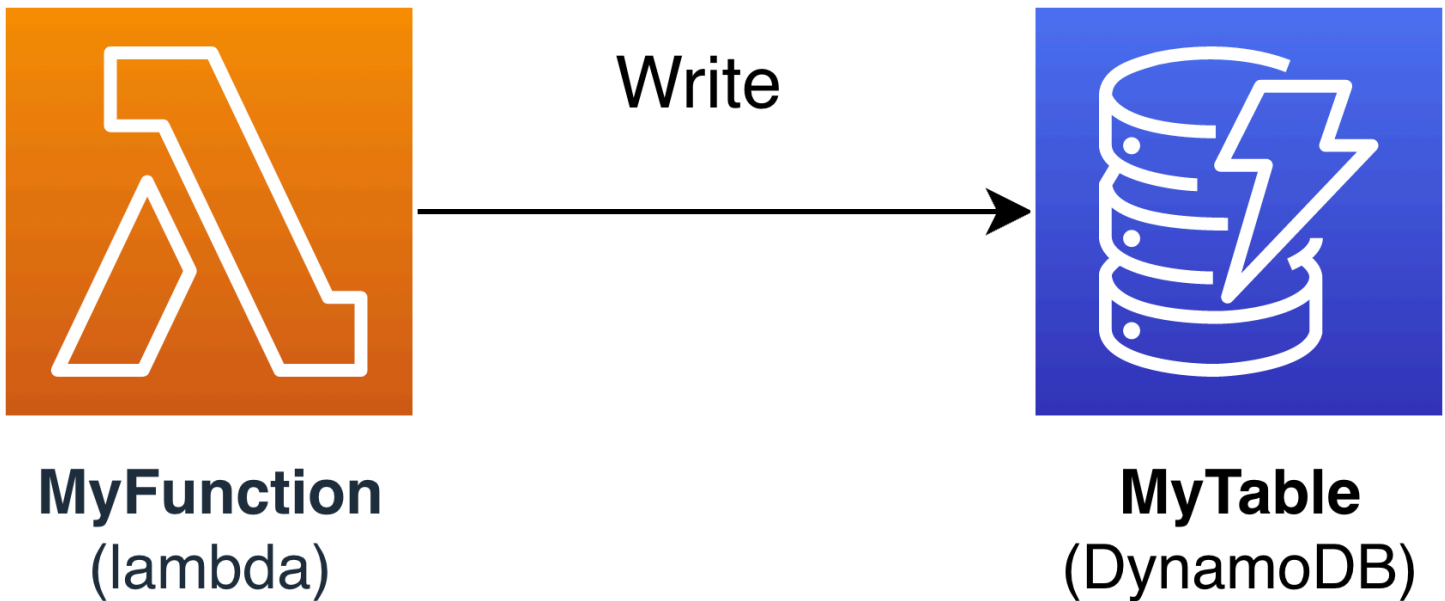
Note

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, consulte [AWS CloudFormation recursos generados cuando se especifica `AWS::Serverless::Connector`](#).

Ejemplo de conectores

El siguiente ejemplo muestra cómo puede utilizar conectores para escribir datos de una AWS Lambda función 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 AWS 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 no sin servidor.

Para obtener una lista de conexiones de recursos compatibles y sus requisitos de propiedad, consulte [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 definan en la misma plantilla, utilice la propiedad `Id`. Si lo desea, puede añadirse un `Qualifier` para reducir el alcance del recurso definido. Cuando el recurso no esté dentro de la misma plantilla, utilice una combinación de propiedades admitidas.

- Para obtener una lista de las combinaciones de propiedades compatibles para los recursos de origen y destino, consulte [Tipos de recursos de origen y destino admitidos para los conectores](#).
- Para obtener una descripción de las propiedades que puede usar con los conectores, consulte [AWS::Serverless::Connector](#).

Cuando defina un recurso de origen con una propiedad distinta de `Id`, utilice 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 `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, puede 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: MyBucket
          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 multidespacho en AWS SAM

Dentro de un recurso de origen, puede 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, consulte la [referencia a los atributos de los recursos](#) en la Guía del usuario de AWS CloudFormation .

Puede 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 temas siguientes:

- [AWS::Serverless::Connector](#)
- [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, [envía un nuevo número](#) al serverless-application-model AWS GitHubrepositorio.

Plantillas de política de AWS SAM

El AWS Serverless Application Model (AWS SAM) le permite elegir de una lista de plantillas de políticas para limitar los permisos de las funciones y máquinas de estado de Lambda a los recursos que utiliza la 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. Utilice 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
```

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: {}
```

Tabla de plantillas de políticas

La siguiente es una tabla de las plantillas de políticas disponibles.

Plantilla de política	Descripción		
AcmGetCertificatePolicy	Otorga permiso para leer un certificado desde. AWS Certificate Manager		
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 AWS CloudFormation pilas.		
CloudWatchDashboardPolicy	Otorga permisos para poner las métricas en los CloudWatch paneles de control para que funcionen.		
CloudWatchAlarmHistoryPolicy	Da permiso para describir el historial CloudWatch de alarmas.		

Plantilla de política	Descripción		
CloudWatchPutMetricsPolicy	Da permiso para enviar métricas a CloudWatch.		
CodeCommitCrudPolicy	Otorga permisos para crear, leer, actualizar o eliminar objetos dentro de un repositorio específico. CodeCommit		
CodeCommitReadPolicy	Otorga permisos para leer objetos dentro de un repositorio específico. CodeCommit		
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.		
DynamoDBCrudPolicy	Concede permisos de creación, lectura, actualización y eliminación a una tabla de Amazon DynamoDB.		

Plantilla de política	Descripción		
DynamoDBReadPolicy	Concede permiso de solo lectura a una tabla DynamoDB.		
DynamoDBReconfigurePolicy	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.		
EKSDescribePolicy	Concede permiso para describir o enumerar los clústeres de Amazon EKS.		

Plantilla de política	Descripción		
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.		
ElasticMapReduceModifyInstanceFleetPolicy	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.		
ElasticsearchHttpPostPolicy	Otorga permiso POST a Amazon OpenSearch Service.		

Plantilla de política	Descripción		
EventBridgePutEventsPolicy	Otorga permisos para enviar eventos a EventBridge.		
FilterLogEventsPolicy	Otorga permiso para filtrar CloudWatch los eventos de registros de un grupo de registros especificado.		
FirehoseCreatePolicy	Otorga permiso para crear, escribir, actualizar y eliminar una transmisión de entrega de Firehose.		
FirehoseWritePolicy	Da permiso para escribir en una transmisión de entrega de Firehose.		
KinesisCreatePolicy	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.		
MobileAnalyticsWriteOnlyAccessPolicy	Concede permiso de solo escritura para poner datos de eventos para todos los recursos de la aplicación.		

Plantilla de política	Descripción		
OrganizationsListAccountsPolicy	Concede permisos de solo lectura para incluir los nombres e ID de las cuentas secundarias.		
PinpointEndpointAccessPolicy	Concede permiso para obtener y actualizar puntos de conexión para una aplicación de Amazon Pinpoint.		
PollyFullAccessPolicy	Concede permiso de acceso completo a los recursos del léxico de Amazon Polly.		
RekognitionDetectOnlyPolicy	Concede permiso para detectar rostros, etiquetas y texto.		
RekognitionFacesManagementPolicy	Concede permiso para añadir, eliminar y buscar rostros en una colección de Amazon Rekognition.		
RekognitionFacesPolicy	Concede permiso para comparar y detectar rostros y etiquetas.		
RekognitionLabelsPolicy	Concede permiso para detectar etiquetas de objetos y de moderación.		
RekognitionNoDataAccessPolicy	Concede permiso para comparar y detectar rostros y etiquetas.		
RekognitionReadPolicy	Concede permiso para enumerar y buscar rostros.		
RekognitionWriteOnAccessPolicy	Concede permiso para crear colecciones e indexar rostros.		

Plantilla de política	Descripción		
Route53ChangeResourceRecordsSetsPolicy	Concede permiso para cambiar los conjuntos de registros de recursos en Route 53.		
S3CrudPolicy	Concede permisos de creación, lectura, actualización y eliminación para actuar en los objetos de un bucket de Amazon S3.		
S3FullAccessPolicy	Concede permiso de acceso completo para actuar sobre los objetos de un bucket de Amazon S3.		
S3ReadPolicy	Concede permiso de solo lectura para leer objetos en un bucket de Amazon Simple Storage Service (Amazon S3).		
S3WritePolicy	Concede permiso de escritura para escribir objetos en un bucket de Amazon S3.		
SageMakerCreateEndpointConfigurationPolicy	Da permiso para crear una configuración de punto final en SageMaker.		
SageMakerCreateEndpointPolicy	Da permiso para crear un punto final en SageMaker.		
ServerlessRepoReadWriteAccessPolicy	Otorga permiso para crear y enumerar aplicaciones en el AWS Serverless Application Repository servicio.		

Plantilla de política	Descripción		
SESBulkTemplatedCrudPolicy	Concede permiso para enviar correos electrónicos, correos electrónicos con plantillas, correos electrónicos masivos con plantillas y verificar la identidad.		
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.		
SESCrudPolicy	Concede permiso para enviar correo electrónico y verificar la identidad.		
SESEmailTemplateCrudPolicy	Concede permiso para crear, obtener, enumerar, actualizar y eliminar plantillas de correo electrónico de Amazon SES.		
SESSendBouncePolicy	Concede SendBounce permiso a una identidad de Amazon Simple Email Service (Amazon SES).		
SNSCrudPolicy	Concede permisos para crear, publicar y suscribirse a temas de Amazon SNS.		
SNSPublishMessagePolicy	Concede permiso para publicar un mensaje en un tema de Amazon Simple Notification Service (Amazon SNS).		
SQSPollerPolicy	Concede permiso para sondear una cola de Amazon Simple Queue Service (Amazon SQS).		
SQSSendMessagePolicy	Concede permisos para enviar mensajes a una cola de Amazon SQS.		

Plantilla de política	Descripción		
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.		
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.		
StepFunctionsExecutionPolicy	Concede permiso para iniciar la ejecución de una máquina de estado de Step Functions.		
TextractDetectAnalyzePolicy	Concede acceso para detectar y analizar documentos con Amazon Textract.		
TextractGetResultPolicy	Concede acceso a documentos detectados y analizados desde Amazon Textract.		
TextractPolicy	Proporciona acceso completo a Amazon Textract.		
VPCAccessPolicy	Permite crear, eliminar, describir y eliminar interfaces de red elásticas.		

Solución 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](#)
- [AWSSecretsManagerGetSecretValuePolicy](#)
- [AWSSecretsManagerRotationPolicy](#)
- [CloudFormationDescribeStacksPolicy](#)
- [CloudWatchDashboardPolicy](#)
- [CloudWatchDescribeAlarmHistoryPolicy](#)
- [CloudWatchPutMetricPolicy](#)

- [CodePipelineLambdaExecutionPolicy](#)
- [CodePipelineReadOnlyPolicy](#)
- [CodeCommitCrudPolicy](#)
- [CodeCommitReadPolicy](#)
- [ComprehendBasicAccessPolicy](#)
- [CostExplorerReadOnlyPolicy](#)
- [DynamoDBBackupFullAccessPolicy](#)
- [DynamoDBCrudPolicy](#)
- [DynamoDBReadPolicy](#)
- [DynamoDBReconfigurePolicy](#)
- [DynamoDBRestoreFromBackupPolicy](#)
- [DynamoDBStreamReadPolicy](#)
- [DynamoDBWritePolicy](#)
- [EC2CopyImagePolicy](#)
- [EC2DescribePolicy](#)
- [EcsRunTaskPolicy](#)
- [EFSWriteAccessPolicy](#)
- [EKSDescribePolicy](#)
- [ElasticMapReduceAddJobFlowStepsPolicy](#)
- [ElasticMapReduceCancelStepsPolicy](#)
- [ElasticMapReduceModifyInstanceFleetPolicy](#)
- [ElasticMapReduceModifyInstanceGroupsPolicy](#)
- [ElasticMapReduceSetTerminationProtectionPolicy](#)
- [ElasticMapReduceTerminateJobFlowsPolicy](#)
- [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 Certificate Manager

```
"Statement": [  
  {  
    "Effect": "Allow",  
    "Action": [  
      "acm:GetCertificate"  
    ],  
    "Resource": {  
      "Fn::Sub": [  
        "${certificateArn}",  
        {  
          "certificateArn": {  
            "Ref": "CertificateArn"  
          }  
        }  
      ]  
    }  
  }  
]
```

AMIDescribePolicy

Da permiso para describir Amazon Machine Images (AMIs).

```
"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"
          }
        }
      ]
    }
  }
]

```

AWSecretsManagerGetSecretValuePolicy

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"
          }
        }
      ]
    }
  }
]

```

AWSecretsManagerRotationPolicy

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 AWS 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 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 permiso de solo lectura al historial de facturación de solo lectura (Cost AWS Cost Explorer Explorer). APIs

```
"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

Da permiso para copiar EC2 imágenes de Amazon.

```

"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "ec2:CopyImage"
    ],
    "Resource": {
      "Fn::Sub": [
        "arn:${AWS::Partition}:ec2:${AWS::Region}:${AWS::AccountId}:image/${imageId}",
        {
          "imageId": {
            "Ref": "ImageId"
          }
        }
      ]
    }
  }
]

```

EC2DescribePolicy

Otorga permiso para describir las instancias de Amazon Elastic Compute Cloud (AmazonEC2).

```

"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

Da permiso para montar un sistema de EFS archivos de Amazon 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

Otorga permiso para describir o enumerar los 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

POSTConcede PUT permiso 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

Otorga permiso para crear, escribir, actualizar y eliminar una transmisión 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

Da permiso para escribir en una transmisión 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

Otorga permiso de solo lectura para incluir los nombres de las cuentas secundarias y. IDs

```

"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"  
          }  
        }  
      ]  
    }  
  }  
]
```

```

        "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

Da permiso para crear una configuración de punto final en SageMaker.

```
"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 SageMaker.

```

"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

Da permiso para enviar correos electrónicos de Amazon, SES correos electrónicos con plantillas y correos masivos con plantillas y para verificar la identidad.

Note

La `ses:SendTemplatedEmail` acción requiere una plantilla. ARN En su lugar, use `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

Da permiso para enviar correos electrónicos de Amazon, SES correos electrónicos con plantillas y correos masivos con plantillas 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

Otorga permiso para crear, obtener, enumerar, actualizar y eliminar plantillas de SES correo electrónico de Amazon.

```

"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "ses:CreateTemplate",
      "ses:GetTemplate",
      "ses:ListTemplates",
      "ses:UpdateTemplate",
      "ses>DeleteTemplate",
      "ses:TestRenderTemplate"
    ],
    "Resource": "*"
  }
]

```

SESSendBouncePolicy

Da SendBounce permiso a una identidad de Amazon Simple Email Service (AmazonSES).

```

"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "ses:SendBounce"
    ],
    "Resource": {
      "Fn::Sub": [
        "arn:${AWS::Partition}:ses:${AWS::Region}:${AWS::AccountId}:identity/
        ${identityName}",
        {
          "identityName": {

```

```

        "Ref": "IdentityName"
      }
    }
  ]
}
]

```

SNSCrudPolicy

Da permiso para crear, publicar y suscribirse a SNS temas de Amazon.

```

"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

Da permiso para publicar un mensaje en un tema de Amazon Simple Notification Service (AmazonSNS).

```

"Statement": [
  {
    "Effect": "Allow",

```

```

"Action": [
  "sns:Publish"
],
"Resource": {
  "Fn::Sub": [
    "arn:${AWS::Partition}:sns:${AWS::Region}:${AWS::AccountId}:${topicName}",
    {
      "topicName": {
        "Ref": "TopicName"
      }
    }
  ]
}
]

```

SQSPollerPolicy

Da permiso para sondear una cola de Amazon Simple Queue Service (AmazonSQS).

```

"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

Da permiso para enviar un mensaje a una SQS cola de Amazon.

```
"Statement": [  
  {  
    "Effect": "Allow",  
    "Action": [  
      "sqs:SendMessage*"  
    ],  
    "Resource": {  
      "Fn::Sub": [  
        "arn:${AWS::Partition}:sqs:${AWS::Region}:${AWS::AccountId}:${queueName}",  
        {  
          "queueName": {  
            "Ref": "QueueName"  
          }  
        }  
      ]  
    }  
  }  
]
```

SSMParameterReadPolicy

Otorga permiso para acceder a un parámetro de un almacén de parámetros de Amazon EC2 Systems Manager (SSM) para cargar datos 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

Otorga permiso para acceder a un parámetro de un almacén de parámetros de Amazon EC2 Systems Manager (SSM) para cargar datos 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 AWS CloudFormation mecanismos

Para controlar el acceso a AWS los recursos, el AWS Serverless Application Model (AWS SAM) puede usar los mismos mecanismos que AWS CloudFormation. Para obtener más información, consulte [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 sobre la concesión de permisos de administrador a un usuario, [consulte Creación del primer usuario y grupo de IAM administradores](#) en la Guía del IAM usuario.

Adjunta las políticas AWS gestionadas 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
- IAMFullAccess
- AWSLambda_FullAccess
- Un mazonAPIGateway administrador
- Amazon S3 FullAccess
- Amazon EC2ContainerRegistryFullAccess

Para obtener información sobre cómo adjuntar políticas a un IAM usuario, consulte [Cambiar los permisos de un IAM usuario](#) en la Guía del IAM usuario.

Otorgue permisos específicos IAM

Para obtener el nivel más detallado de control de acceso, puedes conceder IAM permisos específicos a los usuarios mediante [declaraciones de política](#). Si usa 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 funciones de ejecución de Lambda, consulte [Creación de una función de ejecución en la IAM consola](#).

En el caso [de la aplicación Hello World de ejemplo](#), AWSLambdaBasicExecutionRolebasta 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:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CloudFormationTemplate",
      "Effect": "Allow",
      "Action": [
        "cloudformation:CreateChangeSet"
      ],
      "Resource": [
        "arn:aws:cloudformation:*:aws:transform/Serverless-2016-10-31"
      ]
    },
    {
      "Sid": "CloudFormationStack",
      "Effect": "Allow",
      "Action": [
        "cloudformation:CreateChangeSet",
        "cloudformation:CreateStack",
        "cloudformation>DeleteStack",
        "cloudformation:DescribeChangeSet",
        "cloudformation:DescribeStackEvents",
        "cloudformation:DescribeStacks",
        "cloudformation:ExecuteChangeSet",
        "cloudformation:GetTemplateSummary",
        "cloudformation:ListStackResources",
        "cloudformation:UpdateStack"
      ]
    }
  ]
}
```

```

    ],
    "Resource": [
      "arn:aws:cloudformation:*:111122223333:stack/*"
    ]
  },
  {
    "Sid": "S3",
    "Effect": "Allow",
    "Action": [
      "s3:CreateBucket",
      "s3:GetObject",
      "s3:PutObject"
    ],
    "Resource": [
      "arn:aws:s3:::*/*"
    ]
  },
  {
    "Sid": "ECRRepository",
    "Effect": "Allow",
    "Action": [
      "ecr:BatchCheckLayerAvailability",
      "ecr:BatchGetImage",
      "ecr:CompleteLayerUpload",
      "ecr:CreateRepository",
      "ecr>DeleteRepository",
      "ecr:DescribeImages",
      "ecr:DescribeRepositories",
      "ecr:GetDownloadUrlForLayer",
      "ecr:GetRepositoryPolicy",
      "ecr:InitiateLayerUpload",
      "ecr:ListImages",
      "ecr:PutImage",
      "ecr:SetRepositoryPolicy",
      "ecr:UploadLayerPart"
    ],
    "Resource": [
      "arn:aws:ecr:*:111122223333:repository/*"
    ]
  },
  {
    "Sid": "ECRAuthToken",
    "Effect": "Allow",
    "Action": [

```

```
        "ecr:GetAuthorizationToken"
    ],
    "Resource": [
        "*"
    ]
},
{
    "Sid": "Lambda",
    "Effect": "Allow",
    "Action": [
        "lambda:AddPermission",
        "lambda:CreateFunction",
        "lambda>DeleteFunction",
        "lambda:GetFunction",
        "lambda:GetFunctionConfiguration",
        "lambda:ListTags",
        "lambda:RemovePermission",
        "lambda:TagResource",
        "lambda:UntagResource",
        "lambda:UpdateFunctionCode",
        "lambda:UpdateFunctionConfiguration"
    ],
    "Resource": [
        "arn:aws:lambda:*:111122223333:function:*"
    ]
},
{
    "Sid": "IAM",
    "Effect": "Allow",
    "Action": [
        "iam:CreateRole",
        "iam:AttachRolePolicy",
        "iam>DeleteRole",
        "iam:DetachRolePolicy",
        "iam:GetRole",
        "iam:TagRole"
    ],
    "Resource": [
        "arn:aws:iam:*:111122223333:role/*"
    ]
},
{
    "Sid": "IAMPassRole",
    "Effect": "Allow",
```



```

    "Action": "iam:PassRole",
    "Resource": "*",
    "Condition": {
      "StringEquals": {
        "iam:PassedToService": "lambda.amazonaws.com"
      }
    }
  },
  {
    "Sid": "APIGateway",
    "Effect": "Allow",
    "Action": [
      "apigateway:DELETE",
      "apigateway:GET",
      "apigateway:PATCH",
      "apigateway:POST",
      "apigateway:PUT"
    ],
    "Resource": [
      "arn:aws:apigateway:*:*:*"
    ]
  }
]
}

```

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 sobre IAM las políticas, consulte [Administración de IAM políticas](#) en la Guía del IAM usuario.

Controla el API acceso con tu AWS SAM plantilla

Controlar el acceso a su API Gateway APIs ayuda a garantizar que su aplicación sin servidor sea segura y solo se pueda acceder a ella mediante la autorización que usted habilite. Puede habilitar la autorización en su AWS SAM plantilla para controlar quién puede acceder a su API puerta APIs de enlace.

AWS SAM admite varios mecanismos para controlar el acceso a su API puerta de enlace APIs. 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	✓	✓
IAM permisos		✓
Grupos de usuarios de Amazon Cognito	✓ *	✓
API claves		✓
Políticas de recursos		✓
OAuth2.0/ autorizadores JWT	✓	

* Puede utilizar Amazon Cognito como emisor de JSON Web Token (JWT) con el `AWS::Serverless::HttpApi` tipo de recurso.

- **Autorizadores Lambda:** un autorizador Lambda (anteriormente conocido como autorizador personalizado) es una función de Lambda que se proporciona para controlar el acceso a sus. API Cuando API se llama a usted, esta función 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 sobre los autorizadores Lambda con `AWS::Serverless::HttpApi`, consulte [Trabajar con AWS Lambda autorizadores HTTP APIs en la Guía para desarrolladores](#) de Gateway. API Para obtener más información sobre los autorizadores Lambda con `AWS::Serverless::Api`, consulte Usar autorizadores [API Lambda de Gateway](#) en la Guía para desarrolladores de Gateway. API

Para ver ejemplos de autorizadores de Lambda para cualquier tipo de recurso, consulte [Ejemplos de autorizadores Lambda para AWS SAM](#).

- IAM permisos: [puede controlar quién puede invocar sus API permisos using \(\).AWS Identity and Access Management IAM](#) Los usuarios que te llamen API deben estar autenticados con IAM credenciales. Las llamadas se API realizan correctamente solo si hay una IAM política asociada al IAM usuario que represente a la API persona que llama, un IAM grupo que contenga al usuario o un IAM rol que asuma el usuario.

Solo el tipo `AWS::Serverless::Api` de recurso admite IAM permisos.

Para obtener más información, consulte [Controlar el acceso a un IAM usuario API con permisos](#) en la Guía para desarrolladores de API Gateway. Para ver un ejemplo, consulte [IAM ejemplo de permiso 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 suyo primero API debe iniciar sesión con un usuario en el grupo de usuarios y obtener una identidad o un token de acceso para el usuario. Luego, el cliente te llama API con uno de los tokens devueltos. La API llamada 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 `AWS::Serverless::HttpApi` recurso admite el uso de Amazon Cognito como emisor. JWT

Para obtener más información, consulte [Controlar el acceso a REST API grupos de usuarios de Amazon Cognito como autorizador en la Guía para desarrolladores de API Gateway](#). Para ver un ejemplo, consulte [Ejemplo de grupo de usuarios de Amazon Cognito para AWS SAM](#).

- **API claves:** las API claves son valores de cadena alfanuméricos que usted distribuye a los clientes desarrolladores de aplicaciones para permitir el acceso a las suyas. API

Solo el tipo `AWS::Serverless::Api` de recurso admite API claves.

Para obtener más información sobre API las claves, consulte [Creación y uso de planes de uso con API claves](#) en la Guía para desarrolladores de API Gateway. Para ver un ejemplo de API claves, consulte [API ejemplo clave de AWS SAM](#).

- **Políticas de recursos:** las políticas de recursos son documentos de JSON política que puede adjuntar a un API Gateway API. Utilice las políticas de recursos para controlar si un director específico (normalmente un IAM usuario o un rol) puede invocar el API.

Solo el tipo `AWS::Serverless::Api` de recurso admite las políticas de recursos como mecanismo para controlar el acceso a API Gateway APIs.

Para obtener más información sobre las políticas de recursos, consulte [Controlar el acceso a y API con las políticas de recursos de API Gateway](#) en la Guía para desarrolladores de API Gateway. Para ver ejemplos de políticas de recursos, consulte [Ejemplo de política de recursos para AWS SAM](#).

- **OAuth JWT Autorizadores 2.0/:** puede utilizarlos JWTs como parte de los marcos OpenID [Connect \(OIDC\)](#) y [OAuth2.0](#) para controlar el acceso a sus APIs API Gateway valida las API solicitudes JWTs que los clientes envían y permite o deniega las solicitudes en función de la validación del token y, opcionalmente, del alcance del token.

Solo el tipo de `AWS::Serverless::HttpApi` recurso admite autorizadores OAuth JWT 2.0/.

Para obtener más información, consulte [Controlar el acceso HTTP APIs con JWT autorizadores en la Guía para desarrolladores de API Gateway](#). Para ver un ejemplo, consulte [OAuth Ejemplo de JWT autorizador 2.0/ para AWS SAM](#).

Elegir un mecanismo para controlar el acceso

El mecanismo que elija utilizar para controlar el acceso a su API puerta de enlace APIs 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 su 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 su 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 el mecanismo que desea utilizar, consulte la sección correspondiente en [Ejemplos](#) la sección sobre cómo configurar la aplicación para que utilice ese mecanismo. AWS SAM

Personalizar respuestas de error

Puede usarlo AWS SAM para personalizar el contenido de algunas respuestas de error de API Gateway. Solo el tipo `AWS::Serverless::Api` de recurso admite respuestas personalizadas de API Gateway.

Para obtener más información sobre las respuestas de API Gateway, consulte [las respuestas de API Gateway en Gateway](#) en la Guía para desarrolladores de API Gateway. Para ver un ejemplo de respuestas personalizadas, consulte [Ejemplo de respuesta personalizada para AWS SAM](#).

Ejemplos

- [Ejemplos de autorizadores Lambda para AWS SAM](#)
- [IAMejemplo de permiso para AWS SAM](#)
- [Ejemplo de grupo de usuarios de Amazon Cognito para AWS SAM](#)
- [APIejemplo clave de AWS SAM](#)
- [Ejemplo de política de recursos para AWS SAM](#)
- [OAuthEjemplo de JWT autorizador 2.0/ 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.

Ejemplos de autorizadores **TOKEN** de Lambda (AWS::Serverless::Api)

Puede controlar el acceso al suyo APIs 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, SAM FunctionRole 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 Lambda, consulte [Uso de autorizadores API Lambda de Gateway](#) en la Guía para desarrolladores de Gateway. API

Ejemplos de autorizadores de Lambda **REQUEST** (AWS::Serverless::Api)

Puede controlar el acceso al suyo APIs 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 Lambda, consulte [Uso de autorizadores API Lambda de Gateway](#) en la Guía para desarrolladores de Gateway. API

Ejemplos de autorizadores de Lambda (AWS::Serverless::HttpApi)

Puede controlar el acceso al suyo HTTP APIs 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

```

IAMejemplo de permiso para AWS SAM

Puede controlar el acceso al suyo APIs definiendo IAM los permisos en su AWS SAM plantilla. Para ello, utilice el tipo de datos [ApiAuth](#).

A continuación se muestra un ejemplo de AWS SAM plantilla que se utiliza para IAM los permisos:

```

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 IAM los permisos, consulte [Controlar el acceso para invocar un permiso API](#) en la Guía para desarrolladores de API Gateway.

Ejemplo de grupo de usuarios de Amazon Cognito para AWS SAM

Puede controlar el acceso al suyo APIs definiendo los grupos de usuarios de Amazon Cognito en su AWS SAM plantilla. 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 sobre los grupos de usuarios de Amazon Cognito, consulte [Controlar el acceso a un REST API uso de grupos de usuarios de Amazon Cognito como autorizador en API la Guía para](#) desarrolladores de Gateway.

API ejemplo clave de AWS SAM

Puede controlar el acceso a su plantilla APIs mediante la introducción de API claves en su AWS SAM plantilla. Para ello, utilice el tipo de datos [ApiAuth](#).

A continuación se muestra un ejemplo AWS SAM de sección de plantillas para API claves:

```

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 API las claves, consulte [Creación y uso de planes de uso con API claves](#) en la Guía para desarrolladores de API Gateway.

Ejemplo de política de recursos para AWS SAM

Puede controlar el acceso a la suya APIs adjuntando una política de recursos a su AWS SAM plantilla. Para ello, utilice el tipo de datos [ApiAuth](#).

El siguiente es un ejemplo de AWS SAM plantilla para una plantilla privadaAPI. Una empresa privada API debe tener una política de recursos para poder implementarla.

```

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: '*'
        }
  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, consulte [Controlar el acceso a y API con las políticas de recursos de API Gateway](#) en la Guía para desarrolladores de API Gateway. Para obtener más información sobre la privacidad APIs, consulte [Creación de una red privada API en Amazon API Gateway](#) en la Guía para desarrolladores de API Gateway.

OAuthEjemplo de JWT autorizador 2.0/ para AWS SAM

Puede controlar el acceso a su APIs uso JWTs como parte de los [marcos OpenID Connect \(OIDC\)](#) y [OAuth2.0](#). Para ello, utilice el tipo de datos [HttpApiAuth](#).

La siguiente es una sección de AWS SAM plantilla de ejemplo para un autorizador OAuth 2.0/JWT:

```

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 los JWT autorizadores OAuth 2.0/, consulte [Controlar el acceso HTTP APIs con JWT autorizadores en la Guía para desarrolladores de Gateway. API](#)

Ejemplo de respuesta personalizada para AWS SAM

Puede personalizar algunas respuestas de error de API Gateway definiendo los encabezados de respuesta en su AWS SAM plantilla. Para ello, utilice el tipo de datos [Gateway Response Object](#).

La siguiente es una AWS SAM plantilla de ejemplo 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: "'MyCustomErrorHeader'"
          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 sobre las respuestas de API Gateway, consulte [las respuestas de API Gateway en Gateway](#) en la Guía para desarrolladores de API Gateway.

Aumente la eficiencia con las capas Lambda con AWS SAM

Con AWS SAM, puede incluir capas en sus aplicaciones sin servidor. AWS Lambda las capas permiten extraer código de una función 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 sobre las capas, consulte las [capas Lambda](#) en la Guía para AWS Lambda desarrolladores.

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, consulte [Creación de capas Lambda en AWS SAM](#).

Incluir capas en la aplicación

Para incluir capas en la aplicación, utilice la propiedad `Layers` del tipo de recurso [AWS::Serverless::Function](#).

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</user>
Windows 8	C:\Users\ <user>\AppData\Roaming\AWS SAM</user>
Windows 10	C:\Users\ <user>\AppData\Roaming\AWS SAM</user>
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, ejecute 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, combine 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, combine 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, combine 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 las arquitecturas sin servidor crecen, 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 su aplicación.

Para definir una aplicación anidada en su 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, consulte 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, siga estos pasos:

1. Asegúrese de que ha iniciado sesión en AWS Management Console.
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, consulte [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 su aplicación anidada mediante el comando `aws sam deploy` de la CLI de AWS SAM. Para obtener más información, consulte [Implemente su aplicación y sus recursos con AWS SAM](#).

Note

Cuando implementa una aplicación que contiene aplicaciones anidadas, debe ser conocedor de ello. Para ello, pase `CAPABILITY_AUTO_EXPAND` a la [CreateCloudFormationChangeSet API](#) o utilice el comando. [aws serverlessrepo create-cloud-formation-change-set](#) AWS CLI

Para obtener más información sobre cómo reconocer las aplicaciones anidadas, consulte [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 EventBridge Scheduler 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 particularmente útil para 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 cronológicas 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 de letra muerta () DLQ para 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 a DLQARN, AWS SAM configura los permisos para que el programador programe el envío de mensajes al DLQ Si no DLQ ARN se especifica a, AWS SAM creará el DLQ recurso.

Ejemplos

Ejemplo básico de definición de un evento de EventBridge Scheduler con 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:
    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`.

Organizar los AWS SAM recursos con AWS Step Functions

Puede utilizarlos [AWS Step Functions](#) para organizar AWS Lambda funciones 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 creación de flujos de trabajo complejos y sólidos. Con él [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 añadir transiciones y condiciones donde sea necesario. Esto simplifica el proceso de creación de un flujo de trabajo complejo y sólido.

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 Puede comprobar qué versión tiene ejecutando el comando `aws sam --version`.

Step Functions se basa en los conceptos de [tareas](#) y [máquinas de estado](#). Las máquinas de estado se definen mediante el lenguaje JSON basado en [Amazon States Language](#). 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
- Active las ejecuciones de máquinas de estado con EventBridge eventos de API Gateway o Amazon, según una programación dentro de una AWS SAM plantilla o llamando APIs directamente
- 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. Tenga en cuenta que el archivo `my_state_machine.asl.json` debe estar escrito 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 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 firma de código, consulte [Configuración de la firma de código para funciones de Lambda](#) en la Guía para AWS Lambda desarrolladores.

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. AWS 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: utilice 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 utilice 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 usted proporcione, consulte [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, consulte la [validación de firmas](#) en la Guía para desarrolladores de AWS Lambda .

Ejemplo

Creación de un perfil de firma

Para crearlo, ejecute este comando:

```
aws signer put-signing-profile --platform-id "AWSLambda-SHA384-ECDSA" --profile-name MySigningProfile
```

Si el comando anterior se ejecuta correctamente, verá 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 su código

Puede 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 test-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 test-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 test-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 debe estar habilitado para el bucket de Amazon S3 que utilice 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 usted proporcione, consulte [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 ejecutas el `sam deploy --guided` comando con una aplicación sin servidor configurada con firma de código, te AWS SAM pide que proporciones el perfil de firma que se utilizará para la firma de código. Para obtener más información sobre `sam deploy --guided`, consulte [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 un archivo/ubicación de plantilla diferente con la opción `-t` o `--template`.

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

Después de 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 utilizarlos `sam build` para crear toda su aplicación sin servidor. Además, puedes 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`, consulta los temas de esta sección. Si desea una introducción sobre el uso del comando `sam build`, consulte [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 SAMCLI](#).
- Para ver una lista de opciones de comandos `sam build`, consulte [sam build](#).
- Para ver un ejemplo de uso de `sam build` durante un flujo de trabajo de desarrollo típico, consulte [Paso 2: crear su 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 AWS Lambda función 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](#)
- [Solución 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, consulte [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, consulte [Creación de funciones 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 su 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`, consulte [Crea tu aplicación en AWS SAM](#).
- Para obtener más información sobre cómo configurar estos ajustes en la aplicación, consulte [Compilación predeterminada con AWS SAM](#).

Para compilar una aplicación

1. `cd` a la raíz de su proyecto. Es la misma ubicación que la AWS SAM plantilla.

```
$ cd sam-app
```

2. Ejecute lo siguiente:

```
sam-app $ sam build <arguments> <options>
```


Note

Una opción que se usa con frecuencia es `--use-container`. Para obtener más información, consulte [Creación de una función de Lambda dentro de un contenedor proporcionado](#).

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

```
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 pruebas 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 pruebas locales 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, ejecute `sam build` para actualizar el directorio `.aws-sam` antes de realizar pruebas locales.

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 ejecute `sam build` para actualizar el directorio `.aws-sam/build`.
- Cuando modifique los archivos originales, ejecute `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, ejecute 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, guárdelos en `.aws-sam` para evitar que se sobrescriban.

Opciones para la compilación de Sam

Creación de un solo recurso

Proporcione 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, proporcione 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 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 ver instrucciones, consulte [Instalar Docker](#).

A continuación, se muestra un ejemplo del comando:

```
$ sam build --use-container
```

Puede 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, proporcione 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
```

Transfiera la variables de entorno al contenedor de compilaciones.

Utilice `--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, utilice 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": {
    "GITUB_TOKEN": "TOKEN1"
  },
  "MyFunction2": {
    "GITUB_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, utilice 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, puede 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, consulte [--build-in-source](#).

Solución de problemas

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

Ejemplos

Creación de una aplicación que utilice un tiempo de ejecución nativo y un tipo de paquete `.zip`

Para este ejemplo, consulte [Tutorial: Implemente una aplicación Hello World con AWS SAM](#).

Crear una aplicación que utilice un tiempo de ejecución y un tipo de paquete de imágenes nativas

Primero, ejecute `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, ejecute `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 utilice 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, ejecute `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

SAMCLIDescarga 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`, consulte los enlaces siguientes:

- [Aprendizaje AWS SAM: sam build](#) — Serie «Aprendizaje AWS SAM» de Serverless Land. 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 despliegue \(Sesiones con el SAM S2E8\) — Sesiones](#) con la serie sobre. AWS SAM YouTube
- [AWS SAM compilaciones personalizadas: Cómo usar Makefiles para personalizar compilaciones SAM \(S2E9\) — Sesiones con series en.](#) AWS SAM YouTube

Compilación predeterminada con AWS SAM

Para crear una aplicación sin servidor, utilice el comando `sam build`. 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 propiedad `Layers` de un recurso de función. La propiedad `Layers` contiene una lista de los recursos de [capa de AWS Lambda](#) de los que depende la función 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 empaqueta el código como un archivo `.zip`, debe especificar un tiempo de ejecución de Lambda para su 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, consulte [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](#)
- [Construir funciones fuera de AWS SAM](#)

Creación de un archivo .zip

Para crear una aplicación sin servidor como un archivo `.zip`, declare `PackageType: Zip` para su función sin servidor.

AWS SAM crea su 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, utilice el indicador `--use-container`. Este indicador compila tus funciones de forma local en un contenedor de Docker 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, por ejemplo DockerHub, puede usar la `--build-image` opción y proporcionar el URI de una imagen de contenedor alternativa. A continuación, se muestran dos ejemplos de comandos para crear aplicaciones utilizando imágenes de contenedores del DockerHub repositorio:

```
# Build a Node.js 20 application using a container image pulled from DockerHub
sam build --use-container --build-image amazon/aws-sam-cli-build-image-nodejs20.x

# Build a function resource using the Python 3.12 container image pulled from DockerHub
sam build --use-container --build-image Function1=amazon/aws-sam-cli-build-image-python3.12
```

Para ver una lista de los URI que puede utilizar `--build-image`, consulte la sección [Repositorios de imágenes para AWS SAM](#) que contiene DockerHub los URI correspondientes a varios tiempos de ejecución compatibles.

Para ver ejemplos adicionales de cómo crear una aplicación de archivo .zip, consulte 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, declare `PackageType: Image` para su función sin servidor. También debe 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.

A continuación, se muestra un ejemplo de una sección de atributos de recursos de `Metadata`:

```
Metadata:
  Dockerfile: Dockerfile
  DockerContext: ./hello_world
  DockerTag: v1
```

Para descargar una aplicación de muestra configurada con el tipo de paquete Image, consulte [Tutorial: Implemente una aplicación Hello World con AWS SAM](#) en el tutorial: Implementación de una aplicación Hello World. Cuando aparezca el mensaje que le pregunte qué tipo de paquete desea instalar, elija Image.

Note

Si especificas una imagen base de múltiples arquitecturas 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 utilice la arquitectura de destino específica.

Archivo de variables del entorno del contenedor.

Para proporcionar un archivo JSON que contenga variables de entorno para el contenedor de compilación, use el argumento `--container-env-var-file` junto con el comando `sam build`. Puede 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 proporcione para sus 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": {
    "GITUB_TOKEN": "TOKEN1"
  },
  "MyFunction2": {
    "GITUB_TOKEN": "TOKEN2"
  }
}
```

Guarde 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 proporcione 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, puede 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, consulte [--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 pulled from DockerHub
sam build --use-container --build-image amazon/aws-sam-cli-build-image-nodejs20.x

# Build a function resource using the Python 3.12 container image pulled from DockerHub
sam build --use-container --build-image Function1=amazon/aws-sam-cli-build-image-
python3.12

# 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 crea como una imagen de contenedor:

```
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 caso de las aplicaciones de Node.js, puede utilizar `npm ci` en lugar de `npm install` para instalar dependencias. 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 su 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
```

Construir 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. Puede omitir `sam build` y pasar al siguiente paso de su proceso, como realizar pruebas locales o implementar su 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, su 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 Lambda o capas Lambda específicas. Una función es un recurso que puede invocar para ejecutar el código en Lambda. Una capa Lambda permite extraer código de una función Lambda que luego se puede reutilizar en varias funciones Lambda. Puede optar por personalizar su compilación con funciones 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 Lambda que le ayude a reducir el tamaño de los paquetes de despliegue, separe la lógica de las funciones principales de las dependencias y le permita 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 los tiempos de ejecución de los clientes y la creación de capas 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. Crear solo capas Lambda (en lugar de crear toda la aplicación) puede beneficiarlo de varias maneras. Puede ayudarlo 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 sobre las funciones, consulte los [conceptos de Lambda](#) en la Guía para AWS Lambda desarrolladores.

Temas

- [Creación de funciones Lambda de Node.js con esbuild en AWS SAM](#)
- [Edificio. NETFunciones Lambda con compilación nativa AOT en AWS SAM](#)
- [Creación de funciones de Rust Lambda con in Cargo LambdaAWS SAM](#)
- [Creación de funciones Lambda con tiempos de ejecución personalizados en AWS SAM](#)
- [Creación de capas Lambda en AWS SAM](#)

Creación de funciones Lambda de Node.js con esbuild en AWS SAM

Para compilar y empaquetar AWS Lambda las funciones de Node.js, puede utilizarlas AWS SAMCLI con el paquete JavaScript esbuild. El paquete esbuild admite las funciones Lambda que usted escriba. TypeScript

Para crear una función de Lambda de Node.js con esbuild, añada un objeto de Metadata a su recurso de `AWS::Serverless::Function` y especifique `esbuild` para 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 de Metadata admite las siguientes propiedades.

BuildMethod

Especifica el paquete de URL para su 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 su aplicación.

Externo

Especifica la lista de paquetes a omitir de la compilación. Para más información, consulte [Externo](#) en el sitio web de esbuild.

Formato

Especifica el formato de salida de los JavaScript archivos generados en la aplicación. Para más información, consulte [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 debe minimizar el código de salida agrupado. El valor predeterminado es `true`.

OutExtension

Personalice la extensión de los archivos que genera esbuild. Para más información, consulte [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, el rol de ejecución de la función debe tener permiso para realizar la acción de `kms:Encrypt`.

SourcesContent

Especifica si se debe 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, consulte [Contenido de las fuentes](#) en el sitio web de esbuild.

Destinatario

Especifica la ECMAScript versión de destino. El valor predeterminado es `es2020`.

TypeScript Ejemplo de función Lambda

El siguiente fragmento de AWS SAM plantilla de ejemplo usa esbuild para crear una función Lambda de Node.js a partir del código de 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"  
Sourcemaps: true  
EntryPoints:  
  - app.ts  
External:  
  - "<package-to-exclude>"
```

Edificio. NETFunciones Lambda con compilación nativa AOT en AWS SAM

Cree y empaquete su. NET8 AWS Lambda funciones con AWS Serverless Application Model (AWS SAM), que utilizan la compilación Native Ahead-of-Time (AOT) para mejorar AWS Lambda los tiempos de arranque en frío.

Temas

- [. NET8 AOT Descripción general nativa](#)
- [AWS SAM Utilizándolo con su. NET8 funciones Lambda](#)
- [Requisitos previos de instalación](#)
- [Definir. NET8 funciones Lambda en su plantilla AWS SAM](#)
- [Cree su aplicación con la CLI de AWS SAM.](#)
- [Más información](#)

. NET8 AOT Descripción general nativa

Históricamente,. NETLas funciones Lambda tienen tiempos de arranque en frío que afectan a la experiencia del usuario, a la latencia del sistema y a los costes de uso de las aplicaciones sin servidor. Con. NETAOTCompilación nativa, puede mejorar los tiempos de arranque en frío de sus funciones Lambda. Para obtener más información sobre Native for. AOT NET8, consulte [Uso de Native AOT](#) en el GitHub repositorio de Dotnet.

AWS SAM Utilizándolo con su. NET8 funciones Lambda

Haga lo siguiente para configurar su. NET8 funciones Lambda con AWS Serverless Application Model (AWS SAM):

- Instale los requisitos previos en su máquina de desarrollo.
- Defina. NET8 funciones Lambda en la plantilla. AWS SAM
- Cree su aplicación con. AWS SAMCLI

Requisitos previos de instalación

A continuación, se indican los requisitos previos.

- El AWS SAMCLI
- El .NET Núcleo CLI
- Las herramientas de Amazon.Lambda.Tools. NET Herramienta global básica
- Docker

Instale la CLI AWS SAM

1. Para comprobar si ya tiene la CLI de AWS SAM instalada, ejecute lo siguiente:

```
sam --version
```

2. Para instalar el AWS SAMCLI, consulte [Instale la CLI AWS SAM](#).
3. Para actualizar una versión instalada del AWS SAMCLI, consulte [Actualización de la CLI de AWS SAM](#).

Instale el .NET Núcleo CLI

1. Para descargar e instalar el .NET Core CLI, consulte [Descargar .NET](#) del sitio web de Microsoft.
2. Para obtener más información sobre .NET Núcleo CLI, consulte [.NET Core CLI](#) en la guía para AWS Lambda desarrolladores.

Instale Amazon.Lambda.Tools. NET Herramienta global básica

1. Ejecute el siguiente comando:

```
dotnet tool install -g Amazon.Lambda.Tools
```

2. Si ya tiene la herramienta instalada, puede asegurarse de que es la versión más reciente usando el siguiente comando:

```
dotnet tool update -g Amazon.Lambda.Tools
```

3. Para obtener más información sobre Amazon.Lambda.Tools. NET [La herramienta global básica, consulte las extensiones de .NET CLI](#) repositorio en GitHub.

Instalar Docker

- Si se compila con NativeAOT, Docker es necesario instalarlo. Para obtener las instrucciones de instalación, consulte [Instalación de Docker para usarlo con la CLI de AWS SAM](#).

Definir .NET8 funciones Lambda en su plantilla AWS SAM

Para definir un .NET8 Función Lambda en su AWS SAM plantilla, haga lo siguiente:

1. Ejecute el siguiente comando desde el directorio de inicio de su elección:

```
sam init
```

2. Seleccione esta opción AWS Quick Start Templates para elegir una plantilla de inicio.
3. Elija la plantilla Hello World Example.
4. Introduce la opción de no utilizar el motor de ejecución y el tipo de paquete más popularesn.
5. Para el tiempo de ejecución, elija dotnet8.
6. Para el tipo de paquete, elija Zip.
7. Para tu plantilla de inicio, elige Hello World Example using native AOT.

Instalar Docker

- Para compilar con NativeAOT, Docker es necesario instalarlo. Para obtener las instrucciones de instalación, consulte [Instalación de Docker para usarlo con la CLI de AWS SAM](#).

```
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
```

Cree su aplicación con la CLI de AWS SAM.

Desde el directorio raíz del proyecto, ejecute `sam build` para empezar a crear la aplicación. Si la `PublishAot` propiedad se ha definido en su `NET` Archivo de 8 proyectos, se AWS SAMCLI compilará con una AOT compilación nativa. Para obtener más información sobre la `PublishAot` propiedad, consulte [Native AOT Deployment](#) en Microsoft. NETdocumentación.

Para crear su función, AWS SAMCLI invoca el. NETNúcleo CLI que utiliza Amazon.Lambda.Tools. NETHerramienta global básica.

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 construcción. NET8 funciones Lambda, consulte [Introducción a. NET8 tiempos de ejecución para. AWS Lambda](#)

Para obtener más información sobre el comando `sam build`, consulte [sam build](#).

Creación de funciones de Rust Lambda con in Cargo LambdaAWS SAM

Esta función se encuentra en versión preliminar AWS SAM y está sujeta a cambios.

Utilice la interfaz de línea de AWS Serverless Application Model comandos (AWS SAMCLI) con sus AWS Lambda funciones de Rust.

Temas

- [Requisitos previos](#)

- [Configuración AWS SAM para su uso con las funciones de Rust Lambda](#)
- [Ejemplos](#)

Requisitos previos

Lenguaje Rust

Para instalar Rust, consulte [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, consulte [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, consulte [Instalar Docker](#).

Acceda a la característica beta de la CLI de AWS SAM.

Como esta característica está en versión preliminar, debe acceder usando uno de los siguientes métodos:

1. Use la variables de entorno: `SAM_CLI_BETA_RUST_CARGO_LAMBDA=1`.
2. Añada lo siguiente a su archivo `samconfig.toml`:

```
[default.build.parameters]
beta_features = true
[default.sync.parameters]
beta_features = true
```

3. Utilice la opción `--beta-features` cuando utilice 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 "rust-cargolambda" 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 su uso con las funciones de Rust Lambda

Paso 1: Configura tu AWS SAM plantilla

Configure su AWS SAM plantilla con lo siguiente:

- Binario: opcional. Especifique cuándo su plantilla contiene varias funciones Rust de Lambda.
- BuildMethod – rust-cargolambda.
- CodeUri— ruta a su Cargo.toml archivo.
- Controlador: bootstrap.
- Tiempo de ejecución: provided.al2.

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.

A continuación, se muestra 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.al2  
...
```

Paso 2: use la CLI de AWS SAM con su función Rust de Lambda

Usa cualquier AWS SAMCLI comando con tu AWS SAM plantilla. Para obtener más información, consulte [El AWS SAMCLI](#).

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 - aot.dotnet7 (provided.al2)
  2 - dotnet6
  3 - dotnet5.0
  ...
 18 - python3.7
 19 - python3.10
 20 - ruby2.7
 21 - rust (provided.al2)
Runtime: 21

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.al2)
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.al2
      Architectures:
        - x86_64
      Events:
        HelloWorld:
          Type: Api
          Path: /hello
          Method: get
```

A continuación, ejecute `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 planeas ejecutar el `sam local invoke` comando en macOS, necesitas crear funciones diferentes antes de invocarlo. Para ello, usa el siguiente comando:

- `SAM_BUILD_MODE=debug sam build`

Este comando solo es necesario si se van a realizar pruebas locales. Esto no se recomienda al compilar para el despliegue.

```
hello-rust$ sam build
Starting Build use cache
```

```

Build method "rust-cargolambda" 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-rust/rust_app runtime: provided.al2 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-
samclisourcebucket-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://ggdxec91e9.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 probar, podemos invocar nuestra función Lambda utilizando API el punto final.

```

$ 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/awslabs/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.al2
      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.al2)
Local image was not found.
Removing rapid images for repo public.ecr.aws/sam/emulation-provided.al2
Building
image.....
Using local image: public.ecr.aws/lambda/provided:al2-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 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.al2
...
```

Proyecto de función de Lambda múltiple

Este es un ejemplo de una aplicación sin servidor que contiene varias funciones Rust de Lambda.

Estructura del directorio del proyecto:

```
.
### Cargo.lock
### Cargo.toml
### src
# ### function_a.rs
# ### function_b.rs
### template.yaml
```

AWS SAM plantilla:

```
AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31
...
Resources:
  FunctionA:
    Type: AWS::Serverless::Function
    Metadata:
      BuildMethod: rust-cargolambda
      BuildProperties:
        Binary: function_a
    Properties:
      CodeUri: ./
      Handler: bootstrap
      Runtime: provided.al2
```

```
FunctionB:
  Type: AWS::Serverless::Function
  Metadata:
    BuildMethod: rust-cargolambda
    BuildProperties:
      Binary: function_b
  Properties:
    CodeUri: ./
    Handler: bootstrap
    Runtime: provided.al2
```

Archivo Cargo.toml:

```
[package]
name = "test-handler"
version = "0.1.0"
edition = "2021"

[dependencies]
lambda_runtime = "0.6.0"
serde = "1.0.136"
tokio = { version = "1", features = ["macros"] }
tracing = { version = "0.1", features = ["log"] }
tracing-subscriber = { version = "0.3", default-features = false, features = ["fmt"] }

[[bin]]
name = "function_a"
path = "src/function_a.rs"

[[bin]]
name = "function_b"
path = "src/function_b.rs"
```

Creación de funciones Lambda con tiempos de ejecución personalizados en AWS SAM

Puede usar el comando `sam build` para crear los tiempos de ejecución personalizados necesarios para la función de Lambda. Para declarar que la función de Lambda utilizará 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 debe 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 Rust Lambda con in Cargo LambdaAWS 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 debe 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 debe 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`, consulte la publicación de blog [Tiempo de ejecución de Rust para AWS Lambda](#).

Ejemplo 2: generador de archivos Makefile para Python 3.12 (alternativa al generador incluido)

Es posible que desee utilizar una biblioteca o un módulo que no esté incluido en un paquete de creación. Este ejemplo muestra una AWS SAM plantilla para un entorno de ejecución de Python 3.12 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. Tenga 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 Lambda que luego se puede reutilizar en varias funciones Lambda. Crear solo capas Lambda (en lugar de crear toda la aplicación) puede beneficiarlo de varias maneras. Puede ayudarlo 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, consulte [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 información y ejemplos sobre cómo hacerlo, 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 de Metadata recursos 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 especifica makefile, proporcione 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, puede colocar el código en el subdirectorio python/. Por ejemplo, para NodeJS, puede colocar el código en el subdirectorio nodejs/node_modules/.

Para obtener más información, consulte [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: python3.8
  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, consulte la [ContentUri](#) propiedad del tipo de `AWS::Serverless::LayerVersion` recurso.

Al incluir la sección de atributos del Metadata recurso, puede usar el `sam build` comando para crear la capa, como un objeto independiente o como una dependencia de una AWS Lambda función.

- Como objeto independiente. Es posible que desee crear solo el objeto de la capa, por ejemplo, cuando esté probando localmente un cambio de código en la capa y no necesite compilar toda la aplicación. Para crear la capa de forma independiente, especifique 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 con tiempo de ejecución de Python 3.9

La siguiente AWS SAM plantilla de ejemplo crea una capa en el entorno de ejecución de Python 3.9.

```
Resources:
  MyLayer:
    Type: AWS::Serverless::LayerVersion
    Properties:
```

```

ContentUri: my_layer
CompatibleRuntimes:
  - python3.9
Metadata:
  BuildMethod: python3.9 # Required to have AWS SAM build this layer

```

Ejemplo de plantilla 2: Crear una capa con un archivo Make personalizado

La siguiente AWS SAM plantilla de ejemplo utiliza una capa personalizada `makefile` para crear la capa.

```

Resources:
  MyLayer:
    Type: AWS::Serverless::LayerVersion
    Properties:
      ContentUri: my_layer
      CompatibleRuntimes:
        - python3.8
    Metadata:
      BuildMethod: makefile

```

`makefile` contiene el destino de compilación y los comandos que se ejecutarán. Tenga en cuenta que la propiedad `ContentUri` está establecida en `my_layer`, por lo que el archivo Make debe estar ubicado en la raíz del subdirectorio `my_layer` y el nombre del archivo debe 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"

```

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 crear 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 su aplicación le ayuda a confirmar su funcionalidad, confiabilidad y rendimiento, al mismo tiempo que identifica los problemas (errores) que deberán abordarse.

Esta sección 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 la implementación en la AWS nube. Probar antes de la implementación le ayuda a identificar los problemas de forma proactiva, lo que reduce los costes innecesarios asociados a los problemas de implementación. Cada tema de esta sección describe una prueba que puede realizar, explica las ventajas de utilizarla 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](#)
- [Invoque funciones Lambda localmente con AWS SAM](#)
- [Ejecute API Gateway de forma local con AWS SAM](#)
- [Introducción a las pruebas en la nube con sam remote test-event](#)
- [Introducción a las pruebas en la nube con sam remote invoke](#)
- [Automatice las pruebas de integración local con AWS SAM](#)
- [Genere cargas útiles de eventos de muestra con AWS SAM](#)

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 SAMCLI](#).

Requisitos previos

Para usar `sam local`, complete lo siguiente para instalar la CLI de AWS SAM:

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

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 del comando `sam local`

Use el comando `sam local` con cualquiera de sus subcomandos para realizar diferentes tipos de pruebas locales para su aplicación.

```
$ sam local <subcommand>
```

Para obtener más información acerca de cada subcomando, consulte lo siguiente:

- [Introducción a `sam local generate-event`](#)— Genere Servicio de AWS eventos para las pruebas locales.
- [Introducción a `sam local invoke`](#): inicie una invocación única local de un recurso de función de AWS Lambda .
- [Introducción a `sam local start-api`](#): ejecute 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 `sam local generate-event`

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 AWS SAMCLI, consulte. [¿Qué es el? AWS SAMCLI](#)
- Para ver una lista de opciones de comandos `sam local generate-event`, consulte [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. A continuación, este evento se puede utilizar como fuente de eventos para que una función de AWS Lambda siga procesando los datos.

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`, complete lo siguiente para instalar la CLI de AWS SAM:

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

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 eventos compatibles Servicios de AWS.

Para ver una lista de los compatibles Servicios de AWS

1. Ejecute 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 proporcione 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`, ejecute 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": "EXAMPLE123/5678abcdefghijklambdaisawesome/
mnopqrstuvwxyzABCDEFGH"
      },
      "s3": {
        "s3SchemaVersion": "1.0",
        "configurationId": "testConfigRule",
        "bucket": {
          "name": "example-bucket",
          "ownerIdentity": {
            "principalId": "EXAMPLE"
          },
          "arn": "arn:aws:s3:::example-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, ejecute 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. Utilice 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 MyBucket
```

```
{
  "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": "EXAMPLE123/5678abcdefghijklambdaisawesome/
mnopqrstuvwxyzABCDEFGH"
      },
      "s3": {
        "s3SchemaVersion": "1.0",
        "configurationId": "testConfigRule",
        "bucket": {
          "name": "MyBucket",
          "ownerIdentity": {
            "principalId": "EXAMPLE"
          },
          "arn": "arn:aws:s3:::MyBucket"
        },
        "object": {
          "key": "test/key",
          "size": 1024,
          "eTag": "0123456789abcdef0123456789abcdef",
          "sequencer": "0A1B2C3D4E5F678901"
        }
      }
    }
  ]
}
```

Utilice los eventos generados para realizar pruebas locales

Guarde los eventos generados localmente y utilice otros subcomandos `sam local` para realizar las pruebas.

Para guardar los eventos generados localmente

- Ejecute 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 MyBucket > events/s3.json
```

Para usar un evento generado para realizar pruebas locales

- 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`, consulte [sam local generate-event](#).

Para ver una demostración del uso de `sam local`, consulte [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 invoke`

Utilice el AWS Serverless Application Model `sam local invoke` subcomando Command Line Interface (AWS SAMCLI) para iniciar una única invocación local de una AWS Lambda función.

- Para obtener una introducción a la AWS SAMCLI, consulte. [¿Qué es el? AWS SAMCLI](#)
- Para ver una lista de opciones de comandos `sam local invoke`, consulte [sam local invoke](#).
- Para ver un ejemplo de uso de `sam local invoke` durante un flujo de trabajo de desarrollo típico, consulte [Paso 7: \(opcional\) probar su aplicación localmente](#).

Requisitos previos

Para usar `sam local invoke`, complete lo siguiente para instalar la CLI de AWS SAM:

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

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 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 Lambda de forma local

1. Desde el directorio raíz del proyecto, ejecute lo siguiente:

```
$ sam local invoke <options>
```

2. Si la aplicación contiene más de una función, proporcione 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
```

Puede utilizar estos resultados estándar para automatizar aún más sus procesos de desarrollo local.

Opciones

Transferencia de eventos personalizados para invocar la función de Lambda

Para pasar un evento a la función de Lambda, utilice 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 `sam local generate-event`](#).

Pase variables de entorno al invocar la función de 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
```

Especifique una plantilla o función

Para especificar una plantilla a la que CLI de AWS SAM haga referencia, utilice 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, proporcione 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 Lambda de su proyecto Terraform

Utilice la opción `--hook-name` para probar localmente las funciones de Lambda de sus proyectos Terraform. Para obtener más información, consulte [Uso de la CLI de AWS SAM con Terraform para la depuración y las 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, ejecute `sam local invoke` para probar localmente el código de función actualizado.

Las pruebas locales son una excelente solución para desarrollar y probar rápidamente antes de implementarlas en la nube. Sin embargo, las pruebas locales 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. Le recomendamos que [utilice 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 Lambda localmente

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 Lambda localmente

Esta aplicación tiene una función de Lambda que utiliza una variable de entorno como nombre de tabla de Amazon DynamoDB. A continuación se muestra 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`, consulte [sam local invoke](#).

Para ver una demostración del uso de `sam local`, consulte [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 AWS Lambda las funciones de forma local y realizar pruebas a través de 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 SAMCLI](#).
- Para ver una lista de opciones de comandos `sam local start-api`, consulte [sam local start-api](#).
- Para ver un ejemplo de uso de `sam local start-api` durante un flujo de trabajo de desarrollo típico, consulte [Paso 7: \(opcional\) probar su aplicación localmente](#).

Requisitos previos

Para usar `sam local start-api`, complete lo siguiente para instalar la CLI de AWS SAM:

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

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, ejecute 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. Puede 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, ejecute 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, ejecute su propia solución de compilación para actualizar la función. A continuación, ejecute nuevamente `sam local start-api` para alojar la función.

Funciones 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, consulte [Actualización de la CLI de AWS SAM](#).

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

Reutilice 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. Utilice 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`, ejecute `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.

Especifique una imagen de contenedor para utilizarla en las funciones 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. Utilice 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
```

Puede 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
```

Especifique una plantilla para probarla localmente

Para especificar una plantilla a la que CLI de AWS SAM haga referencia, utilice 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 del host de su función de 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.

Utilice la opción `--container-host` para especificar un host. A continuación, se muestra un ejemplo:

```
$ sam local start-api --container-host host.docker.internal
```

Utilice 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, ejecute `sam local start-api` para probar localmente el código de función actualizado.

Las pruebas locales son una excelente solución para desarrollar y probar rápidamente antes de implementarlas en la nube. Sin embargo, las pruebas locales 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. Le recomendamos que [utilice `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`, consulte [sam local start-api](#).

Introducción a las pruebas con `sam local start-lambda`

Usa el AWS Serverless Application Model `sam local start-lambda` subcomando Command Line Interface (AWS SAMCLI) para invocar tu AWS Lambda función a través de AWS Command Line Interface (AWS CLI) o los SDK. Este comando inicia un punto de conexión local que emula AWS Lambda.

- Para obtener una introducción a la, consulte. [AWS SAMCLI ¿Qué es el? AWS SAMCLI](#)
- Para ver una lista de opciones de comandos `sam local start-lambda`, consulte [sam local start-lambda](#).

Requisitos previos

Para usar `sam local start-lambda`, complete lo siguiente para instalar la CLI de AWS SAM:

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

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, ejecute 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, utilice 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
```

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, ejecute `sam local start-lambda` para probar localmente el código de función actualizado.

Las pruebas locales son una excelente solución para desarrollar y probar rápidamente antes de implementarlas en la nube. Sin embargo, las pruebas locales 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. Le recomendamos que [utilice `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`, consulte [sam local start-lambda](#).

Invoque funciones Lambda localmente con AWS SAM

La invocación local de una función 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 costes innecesarios. Además, las pruebas locales facilitan la depuración.

Puede invocar la función Lambda localmente mediante [sam local invoke](#) el comando y proporcionando el identificador lógico de la función y un archivo de eventos. `sam local invoke` también se acepta `stdin` como evento. Para más información, consulte [Evento](#) en la Guía para desarrolladores de AWS Lambda . Para obtener información sobre los formatos de mensajes de eventos de diferentes AWS servicios, consulte [Uso AWS Lambda con otros servicios](#) en la Guía para AWS Lambda desarrolladores.

Note

El `sam local invoke` comando corresponde al comando AWS Command Line Interface (AWS CLI) [aws lambda invoke](#). Puede utilizar cualquiera de los comandos para invocar una función de Lambda.

Debe 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 archivo JSON que contenga las variables de entorno que desee anular.
2. Use 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, especifique un objeto `Parameters` como el siguiente:

```
{
  "Parameters": {
    "TABLE_NAME": "localtable",
    "BUCKET_NAME": "testBucket",
    "STAGE": "dev"
  }
}
```

Para declarar variables de entorno diferentes para cada recurso, especifique objetos para cada recurso de la siguiente manera:

```
{
  "MyFunction1": {
    "TABLE_NAME": "localtable",
    "BUCKET_NAME": "testBucket",
  },
  "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

Puede 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, utilice el argumento `--env-vars` con los comandos `invoke` o `start-api`. Por ejemplo:

```
sam local invoke --env-vars env.json
```

Capas

Si su aplicación incluye capas, para obtener información sobre cómo depurar los problemas con las capas de su host local, consulte [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 AWS SAM Workshop.

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 costes. Además, la ejecución local facilita 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-cli 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-cli local start-api`, consulte [Introducción a las pruebas con aws-sam-cli local start-api](#).

De forma predeterminada, AWS SAM usa integraciones de AWS Lambda proxy y es compatible con ambos tipos de recursos de API. Para obtener más información sobre las integraciones de proxy para los tipos de recursos de API, consulte [Cómo trabajar con integraciones de AWS Lambda proxy para las API HTTP](#) en la Guía para desarrolladores de API Gateway. Para obtener más información sobre las integraciones de proxy para los tipos de recursos de API, consulte [Cómo trabajar con integraciones de proxy Lambda de API Gateway](#) en la Guía para desarrolladores de API Gateway.

Ejemplo:

```
$ aws-sam-cli local start-api
```

AWS SAM busca automáticamente todas las funciones de la AWS SAM plantilla que tengan definidas las fuentes de eventos de API o las fuentes de eventos. A continuación, monta la función en las rutas HTTP definidas.

En el ejemplo de API siguiente, la función Ratings monta `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 tiene 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, ejecute 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. Use 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, especifique un objeto `Parameters` como el siguiente:

```

{
  "Parameters": {
    "TABLE_NAME": "localtable",

```

```
    "BUCKET_NAME": "testBucket",
    "STAGE": "dev"
  }
}
```

Para declarar variables de entorno diferentes para cada recurso, especifique objetos para cada recurso de la siguiente manera:

```
{
  "MyFunction1": {
    "TABLE_NAME": "localtable",
    "BUCKET_NAME": "testBucket",
  },
  "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

Puede 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, utilice el argumento `--env-vars` con los comandos `invoke` o `start-api`. Por ejemplo:

```
$ sam local start-api --env-vars env.json
```

Capas

Si su aplicación incluye capas, para obtener información sobre cómo depurar los problemas con las capas de su host local, consulte [Aumente la eficiencia con las capas Lambda con AWS SAM](#).

Introducción a las pruebas en la nube con sam remote test-event

Utilice el AWS Serverless Application Model comando Command Line Interface (AWS SAM CLI) `sam remote test-event` para acceder a los eventos de prueba que se pueden compartir para sus AWS Lambda funciones y gestionarlos.

Para obtener más información sobre los eventos de prueba compartibles, consulte [Eventos de prueba compartibles](#) en la Guía para desarrolladores de AWS Lambda .

Temas

- [Configure la CLI de AWS SAM para utilizar sam remote test-event](#).
- [Uso del comando sam remote test-event](#)
- [Uso de eventos de prueba compartibles](#)
- [Gestion de eventos de prueba compartibles](#)

Requisitos previos

Para usar `sam remote test-event`, complete lo siguiente para instalar la CLI de AWS SAM:

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

Si ya lo tiene AWS SAM CLI instalado, le recomendamos que lo actualice a la última versión de la AWS SAM CLI versión. Para obtener más información, consulte [Actualización de la CLI de AWS SAM](#).

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 de la sincronización con Nube de AWS.](#)

Configure la CLI de AWS SAM para utilizar sam remote test-event.

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 y administrar los eventos de prueba de Lambda que se pueden compartir 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, consulte los [Eventos de prueba que se puedan compartir](#) en la Guía para desarrolladores de AWS Lambda .

Uso del comando sam remote test-event

El AWS SAM CLI `sam remote test-event` comando proporciona los siguientes subcomandos que puede utilizar para acceder a los eventos de prueba que se pueden compartir y gestionarlos:

- `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 compartible usando el subcomando `delete` junto con lo siguiente:

- Proporcione el nombre del evento de prueba compartible que se va a eliminar.
- Proporcione 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`, consulte [sam remote test-event delete](#). 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 del registro del EventBridge esquema mediante el `get` subcomando junto con lo siguiente:

- Proporcione el nombre del evento de prueba compartible que se desea obtener.
- Proporcione 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 compartible 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, utilice la opción `--output-file` y proporcione 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`, consulte [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. Utilice el subcomando `list` junto con lo siguiente:

- Proporcione 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 AWS 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`, consulte [sam remote test-event list](#). 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. Utilice el subcomando `put` junto con lo siguiente:

- Proporcione un identificador aceptable de la función de Lambda asociada al evento de prueba compatible.

- Proporcione un nombre para el evento de prueba compartible.
- Proporcione 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`, consulte [sam remote test-event put](#). También puede ejecutar lo siguiente desde: AWS SAM CLI

```
$ sam remote test-event put --help
```

Uso de eventos de prueba compartibles

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 que se puedan compartir a una función de Lambda en la nube](#).

Gestión de eventos de prueba compartibles

Este tema contiene ejemplos sobre cómo administrar y usar los eventos de prueba compartibles.

Obtenga un evento de prueba compartible, 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 compartible: utilice el subcomando `with sam remote test-event get` para recuperar un evento de prueba compartible para una función de Lambda específica y guárdelo localmente:


```
$ sam remote test-event get HelloWorldFunction --stack-name sam-app --name demo-event
--output-file demo-event.json
```

2. Modifique el evento de prueba compartible: utilice un editor de texto de su elección para modificar el evento de prueba compartible.
3. Usar el evento de prueba compartible: use el `sam remote invoke` comando y proporcione 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
```

Obtenga un evento de prueba compartible, 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 compartible directamente a la función de Lambda en la Nube de AWS. A continuación, se muestra un ejemplo:

1. Recuperar el evento de prueba compartible: utilice el subcomando `sam remote test-event get` para recuperar un evento de prueba compartible para una función de Lambda específica y guárdelo localmente:

```
$ sam remote test-event get HelloWorldFunction --stack-name sam-app --name demo-event
--output-file demo-event.json
```

2. Modifique el evento de prueba compartible: utilice un editor de texto de su elección para modificar el evento de prueba compartible.
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 compartible.

```
$ sam remote test-event put HelloWorldFunction --stack-name sam-app --name demo-event
--file demo-event.json --force
```

4. Pase el evento de prueba compartible a la función de Lambda: utilice el comando `sam remote invoke` para pasar el evento de prueba compartible 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 sam remote invoke

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. Puede utilizar `sam remote invoke` para invocar los recursos siguientes:

- Amazon Kinesis Data Streams: envíe registros de datos a las aplicaciones de Kinesis Data Streams.
- AWS Lambda: invoque y pase eventos a sus funciones de Lambda.
- Amazon Simple Queue Service (Amazon SQS): envíe mensajes a colas de Amazon SQS.
- AWS Step Functions: invoque máquinas de estado de Step Functions para iniciar la ejecución.

Para obtener una introducción al AWS SAMCLI, consulte [¿Qué es el? AWS SAMCLI](#).

Para ver un ejemplo de uso de `sam remote invoke` durante un flujo de trabajo de desarrollo típico, consulte [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](#)
- [Configure el archivo de configuración de su proyecto](#)
- [Ejemplos](#)
- [Vínculos relacionados](#)

Requisitos previos

Para usar `sam remote invoke`, complete lo siguiente para instalar la CLI de AWS SAM:

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

También recomendamos actualizar a la última versión de AWS SAMCLI. Para obtener más información, consulte [Actualización de la CLI de AWS SAM](#).

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` de la sincronización con Nube de AWS](#).

Uso del comando `sam remote invoke`

Antes de usar este comando, el recurso debe implementarse en la Nube de AWS.

Use 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, consulte [sam remote invoke](#).

Uso con Kinesis Data Streams

Puede enviar registros de datos a una aplicación de Kinesis Data Streams. AWS SAM CLI enviará su registro de datos y devolverá un ID 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. Proporcione un valor de ID de recurso como argumento para su aplicación Kinesis Data Streams. Para obtener información sobre los ID de recursos válidos, consulte el [ID de recurso](#).
2. Proporcione el registro de datos como un evento para enviarlo a la aplicación Kinesis Data Streams. Puede 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 Lambda

Puede 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 invocará 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 Lambda

1. Proporcione un valor de ID de recurso como argumento para su función de Lambda. Para obtener información sobre los ID de recursos válidos, consulte el [ID de recurso](#).

2. Proporcione un evento para que lo envíe a su función de Lambda. Puede 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 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 utilice `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, consulte [Invocación de una función de Lambda configurada para transmitir respuestas](#).

Transfiera eventos de prueba que se puedan compartir a una función 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, consulte 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 usar el AWS SAM CLI `sam remote test-event` comando para acceder a los eventos de prueba que se pueden compartir y administrarlos. 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 sam remote test-event](#).

Transfiera un evento de prueba que se pueda compartir a una función 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 `--test-event-name` y proporcione 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 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. Proporcione un valor de ID de recurso como argumento para la cola de Amazon SQS. Para obtener información sobre los ID de recursos válidos, consulte el [ID de recurso](#).

- Indique un evento para que lo envíe a la cola de Amazon SQS. Puede 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

Puede 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

- Proporcione 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, consulte el [ID de recurso](#).
- Proporcione un evento para enviarlo a su máquina de estado. Puede 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, consulte [sam remote invoke](#).

Paso de un evento al recurso

Utilice 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\"}}%
```

Configure la salida de respuesta de la CLI de AWS SAM

Al invocar un recurso compatible con `aws-sam-cli` `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": "$LATEST",
    "x-amz-log-result":
"U1RBULQgUmVxdWVzdElkOiAzYmRmOWEzMC03NzZkLTRhOTAtOTRhNi00Y2NjYzBmYzdiNDEgVmVyc2lvbjoyJExBVEVTV
    "x-amzn-trace-id":
"root=1-64908d42-17dab270273fcc6b527dd6b8;sampld=0;lineage=2301f8dc:0"
  },
  "RetryAttempts": 0
},
"StatusCode": 200,
"LogResult":
"U1RBULQgUmVxdWVzdElkOiAzYmRmOWEzMC03NzZkLTRhOTAtOTRhNi00Y2NjYzBmYzdiNDEgVmVyc2lvbjoyJExBVEVTV
"ExecutedVersion": "$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": "$LATEST",
    "x-amz-log-result":
"U1RBULQgUmVxdWVzdElkOiBkMzBkMjgwZi04MTg4LTQzNzItYmM5NC1jZTBmMTYwM2I2YmIgVmVyc2lvbjogJExBVEVTV
    "x-amzn-trace-id":
"root=1-649091fc-771473c7778689627a6122b7;sampled=0;lineage=2301f8dc:0"
  },
  "RetryAttempts": 0
},
"StatusCode": 200,
"LogResult":
"U1RBULQgUmVxdWVzdElkOiBkMzBkMjgwZi04MTg4LTQzNzItYmM5NC1jZTBmMTYwM2I2YmIgVmVyc2lvbjogJExBVEVTV
"ExecutedVersion": "$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 de los parámetros Boto3

Pues `sam remote invoke`, AWS SAM CLI utiliza el AWS SDK para Python (Boto3) para interactuar con sus recursos en la nube. Puede utilizar la opción `--parameter` para personalizar los parámetros Boto3. Para ver una lista de los parámetros admitidos que puede personalizar, consulte [--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"
```

Utilice 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`, consulte [sam remote invoke](#).

Configure el archivo de configuración de su proyecto

Para configurar `sam remote invoke` en su archivo de configuración, utilice `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"
}%
```

Obtenga el ID físico de la aplicación Kinesis Data Streams. A continuación, proporcione 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"
}%
```

Proporcione 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/
ITWg6RPOQcWw8Gn0tNPcEJBEHyVVqboJAPgCritqsvCu",
      "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": "4964625141191480677598090639777867595039988349006774274",
  "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 debe proporcionar el nombre de la AWS 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 puede proporcionar la opción `--stack-name`. A continuación, se muestra un ejemplo:

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


Para invocar una función de Lambda, proporcione el identificador físico como identificador de recurso:

El ID físico se crea cuando se implementa utilizando AWS 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\": {}}"}%
```

Invocación de una función 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. Seleccionamos la plantilla de inicio rápido de Lambda Response Streaming y asignamos un nombre a nuestra 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, consulte [Introducción al uso sam sync de la sincronización 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, ejecute `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_pt15g6_80000gr/T/tmpavrzdhgp.

Execute the following command to deploy the packaged template

```
sam deploy --template-file /var/folders/45/5ct135bx3fn2551_pt15g6_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-
samclisourcebucket-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-	
Transformation	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
creation			
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	AWS::Lambda::Function	StreamingFunction	Resource
creation			
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 ack	lambda-streaming- 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://wxgkcc2dyntgtrwhf2dgdcvy1u0rnnof.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, proporcione su ID físico como ID de recurso:

En primer lugar, use `aws-sam list resources` para obtener el ID físico:

```
$ aws-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, invoque la máquina de estados mediante el ID físico como ID de recurso. Transmita un evento en la línea de comandos con la opción `--event`:

```
$ aws-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:

```
$ aws-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 de la AWS SAM CLI, consulte lo siguiente:

- [sam remote invoke](#)
- [Solución de problemas para AWS SAM CLI](#)

Automatice las pruebas de integración local con AWS SAM

Si bien puede usarse [Introducción a las pruebas con sam local invoke](#) para probar el código manualmente, AWS SAM también le permite probar su código mediante pruebas de integración automatizadas. Las pruebas de integración te ayudan a detectar problemas en las primeras etapas del ciclo de desarrollo, a mejorar la calidad del código y a ahorrar tiempo y, al mismo tiempo, a reducir los costes.

Para crear pruebas de integración automatizadas AWS SAM, primero debe ejecutar pruebas con funciones Lambda locales antes de implementarlas en la AWS nube. El [Introducción a las pruebas con sam local start-lambda](#) comando inicia un punto final local que emula el punto final de invocación de Lambda. Puede invocarlo desde sus pruebas automatizadas. Como este punto final emula el punto final de invocación de Lambda, puede escribir pruebas una vez y luego ejecutarlas (sin ninguna modificación) en la función Lambda local o en una función Lambda implementada. También puede ejecutar las mismas pruebas en una pila de AWS SAM implementada en su canalización de CI/CD.

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 conectarse AWS Lambda en la AWS nube.

Genere cargas útiles de eventos de muestra con AWS SAM

Para probar las funciones de Lambda, puede generar y personalizar ejemplos de cargas útiles de eventos que imiten los datos que recibirán las funciones de Lambda cuando sean activadas por otros servicios. Esto incluye servicios como API Gateway AWS CloudFormation, Amazon S3 y más.

La generación de cargas útiles de eventos de muestra le ayuda a probar el comportamiento de la función 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 AWS de servicio para probar las funciones.

Para ver la lista completa de servicios para los que puede generar cargas útiles de eventos de muestra, utilice este comando:

```
sam local generate-event --help
```

Para ver la lista de opciones que puede usar para un servicio concreto, utilice 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
```

Depure su aplicación sin servidor con AWS SAM

Tras probar la aplicación, estará preparado para depurar cualquier problema que encuentre. 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.

Se puede utilizar AWS SAM para realizar una depuración gradual, que consiste en ejecutar el código una línea o una instrucción a la vez. Al invocar localmente una función 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 las distintas variables y solucionar los problemas del mismo modo que lo haría con cualquier otra aplicación. Puede comprobar si su 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, consulte [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 AWS 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. La depuración gradual de las funciones 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 de realizar una depuración gradual local incluyen establecer puntos de interrupción, inspeccionar variables y ejecutar el código de la función línea por línea. La depuración

gradual local estrecha el ciclo de retroalimentación, ya que le permite encontrar y solucionar los problemas que puedan surgir en la nube.

Puede usar los AWS kits de herramientas para depurar y también puede ejecutar en modo de depuración. AWS SAM Consulte los temas de esta sección para obtener más información.

Uso de kits AWS 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 varios AWS kits de herramientas que funcionan con diferentes combinaciones de IDE y tiempos de ejecución. En la siguiente tabla se enumeran las combinaciones habituales de IDE y tiempo de ejecución 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 habilitado. 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

Se ejecuta 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. [sam local invoke](#)
[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 al 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 quiere cargar un depurador como iKpdb en el tiempo de ejecución de su función de Python, puede 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 iKpdb en tiempo de ejecución con los demás argumentos que haya 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 AWS CloudFormation Linter

AWS CloudFormation Linter (`cfn-lint`) es una herramienta de código abierto que puede utilizar para realizar una validación detallada de sus plantillas. AWS CloudFormation CFN-lint contiene reglas que se guían por la especificación del recurso. AWS 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.AWS 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 sam 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` AWS 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 [AWS 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. El proceso de implementación es una secuencia automatizada de pasos que se llevan a cabo para lanzar una nueva versión de la aplicación sin servidor.

Los temas de esta sección proporcionan orientación sobre las implementaciones automatizadas y manuales. Para implementar la aplicación manualmente, utilice AWS SAMCLI comandos. 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 de 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 usuario de OpenID OIDC Connect () 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 de sistemas y canalizaciones de CI/CD para implementar con AWS SAM](#)
- [Introducción al uso `sam sync` de la sincronización 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 SAMCLI](#).
- Para ver una lista de opciones de comandos `sam deploy`, consulte [sam deploy](#).

- Para ver un ejemplo de uso de `sam deploy` durante un flujo de trabajo de desarrollo típico, consulte [Paso 3: Implemente la aplicación en el 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`, complete lo siguiente para instalar la CLI de AWS SAM:

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

Antes de usar `sam deploy`, se recomienda tener conocimientos básicos de lo siguiente:

- [Configuración de la CLI de AWS SAM](#).
- [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, utilice 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. Ejecute 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.

Proporcione 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, utilice `sam deploy --guided` o modifique directamente los archivos de configuración.

El siguiente es un ejemplo de salida:

```
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 funciones empaquetadas como un archivo.zip, comprime y carga el AWS SAMCLI paquete en un depósito 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 funciones de Lambda empaquetadas como una imagen de contenedor, AWS SAMCLI 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 AWS CloudFormation como una pila.
 - 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-
samclisourcebucket-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-
samclisourcebucket-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-zip	-

 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	API Gateway endpoint URL for Prod stage for Hello World function
Value	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 AWS CloudFormation consola directamente con `https://console.aws.amazon.com/cloudformation`. URL](https://console.aws.amazon.com/cloudformation)
2. Seleccione Stacks (Pilas).
3. Identifique su pila por el nombre de la aplicación y selecciónela.

Compruebe los cambios antes de la implementación

Puede configurarlo AWS SAMCLI para que muestre el conjunto de AWS 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
```

Especifique parámetros adicionales durante la implementación

Puede 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 salida:

```
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 las funciones 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 salida:

```
#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 `aws-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 `aws-sam build` para actualizar el directorio `.aws-sam` antes de implementarla.
- Al implementar una aplicación por primera vez, utilícela con `aws-sam deploy --guided` para configurar los ajustes de implementación. Para las implementaciones posteriores, puede utilizar `aws-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 `aws-sam deploy`. Para obtener una lista de todas las opciones, consulte [sam deploy](#).

Utilice el flujo interactivo guiado para implementar la aplicación

Utilice la opción `--guided` para configurar los ajustes de implementación de la aplicación mediante un flujo interactivo. A continuación, se muestra un ejemplo:

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

La configuración de implementación de la aplicación se guarda en el archivo `samconfig.toml` del proyecto. Para obtener más información, consulte [Configurar los ajustes del proyecto](#).

Resolución de problemas

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

Ejemplos

Implemente una aplicación Hello World que contenga una función de Lambda empaquetada como un archivo.zip

Para ver un ejemplo, consulte el tutorial [Paso 3: Implemente la aplicación en el Nube de AWS](#) de la aplicación Hello World.

Implemente una aplicación Hello World que contenga una función de Lambda empaquetada como 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, ejecute `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-samclisourcebucket-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-
samclisourcebucket-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          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-deploy1680634124/0ffd4faf-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  AWS::CloudFormation::S  sam-app              User
Initiated
                    tack

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 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 tage	-

CREATE_IN_PROGRESS creation	AWS::ApiGateway::Stage	ServerlessRestApiProdS tage	Resource Initiated
CREATE_COMPLETE	AWS::ApiGateway::Stage	ServerlessRestApiProdS tage	-
CREATE_COMPLETE	AWS::Lambda::Permissio n	HelloWorldFunctionHell oWorldPermissionProd	-
CREATE_COMPLETE	AWS::CloudFormation::S tack	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- 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 AWS SAM, puede implementar su aplicación manualmente y también puede automatizar las implementaciones. Úselo AWS SAMCLI para implementar su aplicación manualmente. Para automatizar la implementación, utilice canalizaciones y un sistema de integración e implementación continuas (CI/CD). Los temas de esta sección proporcionan información sobre ambos enfoques.

Temas

- [Cómo utilizarlos para realizar AWS SAMCLI una implementación manual](#)
- [Implemente con sistemas y canalizaciones de CI/CD](#)
- [Implementación gradual](#)
- [Solución de problemas de las implementaciones mediante la CLI de AWS SAM](#)
- [Más información](#)

Cómo utilizarlos para realizar AWS SAMCLI una implementación manual

Después de desarrollar y probar la aplicación sin servidor a nivel local, puede 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 sistemas y canalizaciones de CI/CD

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 tareas de CI/CD para aplicaciones sin servidor. Varios sistemas de CI/CD admiten la AWS SAM creación de imágenes de contenedores y AWS SAM también proporcionan un conjunto de plantillas de canalización predeterminadas para varios sistemas de CI/CD que resumen las mejores prácticas de implementación. AWS

Para obtener más información, consulte [Uso de sistemas y canalizaciones de CI/CD 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 proporcionan. AWS CodeDeploy 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 gradualmente, 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 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, dispone de las siguientes opciones:

- Configure su aplicación con autorización. Para obtener más información acerca de la autorización, consulte [Controla el API acceso con tu AWS SAM plantilla](#).
- Responda a esta pregunta **Y** para indicar que está 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 de sistemas y canalizaciones de CI/CD para implementar con AWS SAM

AWS SAM ayuda a las organizaciones a crear canales para sus sistemas de CI/CD preferidos, de modo que puedan aprovechar los beneficios de la CI/CD con un esfuerzo mínimo, como acelerar la frecuencia de implementación, acortar el tiempo de espera de los cambios y reducir los errores de implementación.

AWS SAM simplifica las tareas de CI/CD para aplicaciones sin servidor mediante 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 SAM CLI Estas imágenes también reducen la necesidad de que los equipos creen y administren sus propias imágenes para los sistemas de CI/CD. Para obtener más información sobre la AWS SAM creación de imágenes de contenedores, consulte [Repositorios de imágenes para AWS SAM](#).

Varios sistemas de CI/CD admiten la AWS SAM creación de imágenes de contenedores. El sistema de CI/CD 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 sistemas de CI/CD que resumen las mejores prácticas de implementación. AWS Estas plantillas de canalización predeterminadas utilizan formatos de configuración de canalización JSON/YAML estándar, y las prácticas recomendadas integradas ayudan a realizar implementaciones con varias cuentas y regiones, y a comprobar 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 SAM CLI comandos o 2) Generar un ejemplo de configuración de canalización de CI/CD que pueda usar como punto de partida para su propia aplicación.

Temas

- [¿Qué es una canalización?](#)
- [Genere una canalización de CI/CD 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](#)
- [Cómo cargar archivos locales durante la implementación con AWS SAM CLI](#)

¿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 SAM, puede usar muchos de los sistemas de CI/CD más comunes para implementar sus aplicaciones, incluidos Jenkins AWS CodePipeline, GitLab CI/CD y Actions. GitHub](#)

Las plantillas de Pipeline incluyen las mejores prácticas AWS de implementación para facilitar las implementaciones con varias cuentas y regiones. AWS Los entornos como los de desarrollo y producción suelen existir en cuentas diferentes. AWS Esto permite a los equipos de desarrollo configurar canales de despliegue seguros, 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.

Genere una canalización de CI/CD 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 sistema de CI/CD que elija usar. Su proceso de implementación es lo que configura y usa para automatizar la implementación de su aplicación sin servidor. Hay una plantilla de canalización inicial preconfigurada para ayudarle a configurar rápidamente la canalización de despliegue de su aplicación sin servidor.

Con una plantilla de canalización inicial, puedes generar canalizaciones en cuestión de minutos con el comando. [sam pipeline init](#)

Las plantillas de canalización iniciales utilizan la conocida YAML sintaxisJSON/del sistema CI/CD 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, AWS SAM CLI permite generar configuraciones de canalización de CI/CD iniciales para las canalizaciones de CI/CD de Jenkins AWS CodePipeline, GitLab CI/CD, Actions y Bitbucket.](#) [GitHub](#)

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 IAM usuario y las funciones con los permisos necesarios, un bucket de Amazon S3 y, opcionalmente, un ECR repositorio de Amazon.
2. Conecta tu repositorio de Git con tu sistema de CI/CD: tu sistema de CI/CD 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 de Git y sistema de CI/CD 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 tu canalización en tu repositorio de Git: este paso es necesario para garantizar que tu sistema de CI/CD 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 ejecute automáticamente.

El orden de estos pasos y los detalles de cada uno de ellos varían en función del sistema de CI/CD:

- Si lo 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, lleve a cabo las siguientes tareas en este orden:

1. Cree recursos de infraestructura
2. Genere la configuración de la canalización
3. Confirme la configuración de su canalización en Git
4. Conecte su repositorio de Git con su sistema CI/CD

Note

El siguiente procedimiento utiliza 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 IAM los usuarios y las funciones) 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 IAM usuario y las funciones con los permisos necesarios, un bucket de Amazon S3 y, opcionalmente, un ECR repositorio de Amazon. Debe tener un conjunto de recursos de infraestructura para cada etapa de implementación de la canalización.

Puede ejecutar el siguiente comando para editar o agregar esta configuración:

```
sam pipeline bootstrap
```

Note

Ejecute 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, ejecute 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 su sistema de CI/CD conozca la configuración de la canalización y se ejecute cuando se realicen los cambios.

Paso 4: conecte su repositorio de Git con su sistema CI/CD

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 desea utilizar una ramificación que no sea main, debe volver a ejecutar el comando `sam deploy`. Tenga en cuenta que, en función del repositorio de Git que utilice, es posible que también tenga 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 canalización de CI/CD, consulte [CI/CD](#) en The Complete Workshop. AWS CodePipeline 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. Cree recursos de infraestructura
2. Conecte su repositorio de Git con su sistema CI/CD
3. Cree objetos de credenciales
4. Genere la configuración de la canalización
5. Confirme su configuración de la canalización en el repositorio de Git

Note

El siguiente procedimiento utiliza 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 IAM los usuarios y las

funciones) tienen más permisos que los desarrolladores (es decir, los usuarios que solo necesitan permiso para configurar canalizaciones individuales, pero no los recursos de infraestructura necesarios). AWS

Paso 1: cree recursos de infraestructura

Las canalizaciones que se utilizan AWS SAM requieren determinados AWS recursos, como un IAM usuario y las funciones con los permisos necesarios, un bucket de Amazon S3 y, opcionalmente, un ECR repositorio de Amazon. Debe tener un conjunto de recursos de infraestructura para cada etapa de implementación de la canalización.

Puede ejecutar el siguiente comando para editar o agregar esta configuración:

```
sam pipeline bootstrap
```

Note

Ejecute 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: conecte su repositorio de Git con su sistema CI/CD

Es necesario conectar el repositorio de Git al sistema de CI/CD para que el sistema de CI/CD pueda acceder al código fuente de la aplicación para creaciones e implementaciones.

Note

Puede omitir este paso si utiliza una de las siguientes combinaciones, ya que la conexión se realiza automáticamente:

1. GitHub Acciones con el GitHub repositorio
2. GitLab CI/CD con repositorio GitLab
3. Bitbucket Pipelines con repositorio de Bitbucket

Para conectar el repositorio de Git con el sistema de CI/CD, realice una de las siguientes acciones:

- Si utiliza Jenkins, consulte 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 sistema CI/CD tiene su propia forma de gestionar las credenciales necesarias para que el sistema CI/CD acceda a su repositorio de Git.

Para crear los objetos de credenciales necesarios, realice una de las siguientes acciones:

- Si utiliza Jenkins, cree una “credencial” única que almacene tanto el identificador como la contraseña. Siga las instrucciones del blog [Cómo crear una canalización con Jenkins con AWS SAM](#), en la sección Configurar Jenkins. Necesitará 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 utiliza Bitbucket Pipelines, cree dos “variables seguras”, una para cada identificador y contraseña. Siga las instrucciones de las [variables y secretos](#); necesitará dos “nombres secretos” para el siguiente paso.

Paso 4: genere la configuración de la canalización

Para generar la configuración de la canalización, ejecute el siguiente comando. Necesitará introducir el objeto de credenciales que creó en el paso anterior:

```
sam pipeline init
```

Paso 5: confirme la configuración de su canalización en el repositorio de Git

Este paso es necesario para garantizar que su sistema de CI/CD 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 canalización de CI/CD usando GitHub Actions, consulte [CI/CD con GitHub](#) en El taller completo de AWS SAM .

Cómo personalizar las canalizaciones iniciales con AWS SAM

Como administrador de CI/CD, es posible que desee personalizar una plantilla de canalización inicial, y las indicaciones guiadas asociadas, que los desarrolladores de su empresa puedan utilizar 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 puede 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. Cree un archivo de **questions.json**: el archivo `questions.json` debe 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`, consulte [questions.json.schema](#). Para ver un ejemplo de archivo `questions.json`, consulte [questions.json](#).
2. Asigne las claves de preguntas a los nombres de los cookiecutter: cada objeto del archivo `questions.json` necesita una clave 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, consulte la sección [Archivos de ejemplo](#) más adelante en este tema.
3. Cree 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 debe solicitar información o, en el caso de la opción `--bootstrap`, para cuántas etapas debe crear los recursos de infraestructura. Para ver un ejemplo de archivo `metadata.json` que declara una canalización con dos etapas, consulte [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, consulte 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 sistema de CI/CD que utilice. Por este motivo, los ejemplos de esta sección muestran cómo configurar varios sistemas de CI/CD para automatizar la creación de aplicaciones sin servidor en una imagen de contenedor de compilación. Estas imágenes de contenedores de compilación facilitan la creación y el empaquetado de aplicaciones sin servidor mediante el AWS SAM CLI.

Los procedimientos para que su canalización CI/CD existente implemente aplicaciones sin servidor utilizando AWS SAM son ligeramente diferentes en función del sistema CI/CD que utilice.

En los temas siguientes se proporcionan ejemplos para configurar el sistema de CI/CD para crear aplicaciones sin servidor dentro de una AWS SAM imagen de contenedor de 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 AWS 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. Configura las etapas de la 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`.

AWS 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 las imágenes de contenedores de compilación de Amazon Elastic Container Registry (Amazon ECR) disponibles para diferentes tiempos de ejecución, consulte [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. Configura las etapas del proceso 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 las imágenes de contenedores de compilación de Amazon Elastic Container Registry (Amazon ECR) disponibles para diferentes tiempos de ejecución, consulte [Repositorios de imágenes para AWS SAM](#).

Uso de Jenkins para implementar con AWS SAM

Para configurar su canalización de [Jenkins](#) para automatizar la creación y el despliegue de su AWS SAM aplicación, `Jenkinsfile` 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. Configura las etapas de la 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 las imágenes de contenedores de compilación de Amazon Elastic Container Registry (Amazon ECR) disponibles para diferentes tiempos de ejecución, consulte [Repositorios de imágenes para AWS SAM](#).

Uso de GitLab CI/CD para implementar con AWS SAM

Para configurar la [GitLab](#) canalización para automatizar 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. Configura las etapas de la 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 las imágenes de contenedores de compilación de Amazon Elastic Container Registry (Amazon ECR) disponibles para diferentes tiempos de ejecución, consulte [Repositorios de imágenes para AWS SAM](#).

Uso de GitHub acciones para implementar con AWS SAM

Para configurar su [GitHub](#) proceso de forma que automatice la creación y el despliegue de su AWS SAM aplicación, primero debe instalar la interfaz de línea de AWS SAM comandos (CLI) en su 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 SAM CLI 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 las imágenes de contenedores de compilación de Amazon Elastic Container Registry (Amazon ECR) disponibles para diferentes tiempos de ejecución, consulte [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 plataformas de integración GitLab continua y entrega continua (CI/CD). Con este soporte, puede usar cuentas de usuario de CI/CD autorizadas de cualquiera de estas plataformas para administrar sus 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 le pida que elija un proveedor de identidad, seleccione OIDC.
2. A continuación, seleccione un proveedor de OIDC compatible.
3. Introduzca 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 plataforma CI/CD 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. Introduzca el número de una configuración para editarlo o pulse `Enter` para continuar.
6. Cuando se le pida que confirme la creación de los recursos necesarios para admitir la conexión OIDC introducida, pulse `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 recurso AWS CloudFormation, consulte [AWS::IAM::OIDCProvider](#) en la Guía del usuario de AWS CloudFormation.

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](#)

Cómo cargar archivos locales durante la implementación con AWS SAMCLI

Al desarrollar, a menudo le resultará beneficioso dividir el código de la aplicación en archivos separados para organizar y administrar mejor la aplicación. Un ejemplo básico de esto es separar el código de AWS Lambda función del código de infraestructura. Para ello, organiza el código de la función Lambda en un subdirectorio del proyecto y hace referencia a su ruta local dentro de la plantilla (`template.yaml`). AWS Serverless Application Model AWS SAM

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). Puede utilizar la CLI de AWS SAM para facilitar este proceso de forma automática. Utilice el comando `sam deploy` o `sam package` para hacer lo siguiente:

1. Cargue automáticamente sus archivos locales a un AWS servicio accesible.
2. Actualice automáticamente la plantilla de su solicitud para hacer referencia a la nueva ruta del archivo.

Temas

- [Demostración: utilice la CLI de AWS SAM para cargar el código de la función de Lambda](#)
- [Casos de uso admitidos](#)
- [Más información](#)

Demostración: utilice la CLI de AWS SAM para cargar el código de la función de Lambda

En esta demostración, inicializamos la aplicación Hello World de ejemplo con un tipo de paquete .zip para la función de 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, ejecute `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, ejecute `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-
samclisourcebucket-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 AWS 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-samclisourcebucket-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 AWS CloudFormation recursos y AWS CloudFormation macros.

Tipos de archivos

Se admiten archivos de aplicaciones e imágenes de Docker.

AWS 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

Recurso	Propiedades
AWS::AppSync::FunctionConfiguration	CodeS3Location RequestMappingTemplateS3Location ResponseMappingTemplateS3Location
AWS::AppSync::GraphQLSchema	DefinitionS3Location
AWS::AppSync::Resolver	CodeS3Location RequestMappingTemplateS3Location ResponseMappingTemplateS3Location
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

Recurso	Propiedades
<code>AWS::Serverless::Function</code>	CodeUri ImageUri
<code>AWS::Serverless::GraphQLApi</code>	SchemaUri Function.CodeUri Resolver.CodeUri
<code>AWS::Serverless::HttpApi</code>	DefinitionUri
<code>AWS::Serverless::LayerVersion</code>	ContentUri
<code>AWS::Serverless::StateMachine</code>	DefinitionUri
<code>AWS::StepFunctions::StateMachine</code>	DefinitionS3Location

AWS 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.

Introducción al uso `sam sync` de la sincronización 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. Utilice `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 su 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 usar `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 su 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 su aplicación con la Nube de AWS. Esto hace lo siguiente:

1. Crear su aplicación: este proceso es similar a usar el comando `sam build`.
2. Implementar su aplicación: la CLI de AWS SAM implementa su 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 su aplicación. Esto es lo que ofrece la opción `--watch`.

Esta opción puede que esté activada de forma predeterminada. Para ver los valores predeterminados, consulte el archivo `samconfig.toml` de su 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 AWS 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 ejecute `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 AWS Lambda función. 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, utilice la opción `--resource-id`. A continuación, se muestra un ejemplo:

```
$ sam sync --code --resource-id HelloWorldFunction --resource-id HelloWorldLayer
```

Preparar su 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, consulte [sam sync](#).

Realice un AWS CloudFormation despliegue único

Use la opción `--no-watch` para desactivar la sincronización automática. A continuación, se muestra un ejemplo:

```
$ sam sync --no-watch
```

AWS SAMCLI realizará un AWS CloudFormation despliegue único. Este comando agrupa las acciones realizadas por los comandos `sam build` y `sam deploy`.

Omita la AWS CloudFormation implementación inicial

Puede personalizar si se requiere una AWS CloudFormation implementación cada vez que `sam sync` se ejecute.

- Asegúrese `--no-skip-deploy-sync` de requerir una AWS CloudFormation implementación cada vez `sam sync` que se ejecute. Esto garantiza que su infraestructura local esté sincronizada AWS 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 AWS CloudFormation implementación sea opcional. AWS SAMCLI comparará su AWS SAM plantilla local con la AWS CloudFormation plantilla implementada y omitirá la AWS CloudFormation implementación inicial si no se detecta ningún cambio. Omitir la AWS 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 AWS CloudFormation despliegue en los siguientes escenarios:

- Si han pasado 7 días o más desde la última AWS CloudFormation implementación.
- Si se detecta una gran cantidad de cambios en el código de la función Lambda, la AWS 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. Proporcione 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, consulte [Reutilice el código y los recursos mediante aplicaciones anidadas en AWS SAM](#).

Especifique una AWS CloudFormation pila específica para actualizar

Para especificar una AWS CloudFormation pila específica para actualizarla, proporciona 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, puede 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, consulte [--build-in-source](#).

Especificar archivos y carpetas que no iniciarán una sincronización

Utilice 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, consulte [--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 [Solución de problemas para AWS SAM CLI](#).

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, consulte [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_pt15g6_80000gr/T/tmpx_5t4u3f.
```

```
Execute the following command to deploy the packaged template
```

```
sam deploy --template-file /var/folders/45/5ct135bx3fn2551_pt15g6_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-
samclisourcebucket-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	ssionProd
-		ssionProd

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-2P1N6TPTQoco
-----
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 SAM CLI detecta 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 AWS CloudFormation implementación AWS SAMCLI automáticamente. El siguiente es un ejemplo de salida:

```

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-
samclisourcebucket-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           ResourceType
LogicalResourceId       ResourceStatusReason
-----

```

UPDATE_IN_PROGRESS	AWS::CloudFormation::Stack	sam-app
Transformation succeeded		
UPDATE_IN_PROGRESS	AWS::CloudFormation::Stack	
AwsSamAutoDependencyLayerNestedSt	-	ack
UPDATE_COMPLETE	AWS::CloudFormation::Stack	
AwsSamAutoDependencyLayerNestedSt	-	ack
UPDATE_IN_PROGRESS	AWS::ApiGateway::RestApi	
ServerlessRestApi	-	
UPDATE_COMPLETE	AWS::ApiGateway::RestApi	
ServerlessRestApi	-	
CREATE_IN_PROGRESS	AWS::ApiGateway::Deployment	
ServerlessRestApiDeployment8cf30e	-	d3cd
UPDATE_IN_PROGRESS	AWS::Lambda::Permission	
HelloWorldFunctionHelloWorldPermi	Requested update requires the	ssionProd
	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	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
  -                                                                    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-2P1N6TPTQoco
-----

```

Stack update succeeded. Sync infra completed.

Infra sync completed.

Más información

Para ver una descripción de las opciones de `aws sam sync`, consulte [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](#)
- [Siguiendo pasos](#)

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.

Configure mediante la CLI de AWS SAM

Al inicializar su aplicación con `aws 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 SAM CLI 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.

```
aws sam init --application-insights
```

Para obtener más información sobre el uso del comando `aws sam init`, consulte [aws 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, consulte [AWS::ResourceGroups::Group](#) en la Guía del usuario de AWS CloudFormation .
- `AWS::ApplicationInsights::Application`— Configura CloudWatch Application Insights para el grupo de recursos. Para obtener más información sobre este tipo de recurso, consulte [AWS::ApplicationInsights::Application](#) en la Guía del usuario de AWS CloudFormation .

Ambos recursos se transfieren automáticamente AWS CloudFormation al implementar la aplicación. Puede usar la AWS CloudFormation sintaxis de la AWS SAM plantilla para configurar aún más CloudWatch Application Insights. Para obtener más información, consulta [Uso de AWS 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
```

Siguientes pasos

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 `sam logs`. Este comando le permite obtener los registros generados por la función de Lambda desde la línea de comandos.

Note

El `sam logs` comando funciona para todas AWS Lambda las funciones, no solo para las que se despliegan AWS SAM.

Obtención de registros por pila AWS CloudFormation

Cuando tu función forma parte de una AWS 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 Lambda

O bien, puede obtener registros utilizando el nombre de la función:

```
sam logs -n mystack-HelloWorldFunction-1FJ8PD
```

Rastreo de registros

Agregue 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

Puede 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

Utilice 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.

JSONbonita impresión

Si sus mensajes de registro imprimen JSON cadenas, AWS SAMCLI automáticamente las imprime de forma bonita JSON para ayudarlo a analizarlas y comprenderlas visualmenteJSON.

AWS SAM referencia

Esta sección contiene material AWS SAM de referencia. Incluye material de AWS SAMCLI referencia, como información de referencia sobre AWS SAMCLI los comandos, e AWS SAMCLI información adicional, 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 la plantilla AWS SAM

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. [El AWS SAM proyecto y la 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 los AWS recursos. La AWS SAM plantilla es una extensión de AWS CloudFormation la plantilla. Para obtener una referencia completa sobre las plantillas de AWS CloudFormation , consulte la [Referencia de plantillas de](#) en la Guía del usuario de AWS CloudFormation .

Referencia de los comandos AWS SAM CLI

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 aplicaciones sin servidor.

Puede 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 utiliza la CLI de AWS SAM, puede 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.

- `aws-sam local invoke` y `aws-sam local start-api`: utilice estos comandos para probar el código de la aplicación de forma local antes de implementarlo en Nube de AWS.
- `aws-sam logs`: utilice este comando para recuperar los registros que genera la función de Lambda. Esto puede ayudarlo a probar y depurar la aplicación después de haberla implementado en Nube de AWS.
- `aws-sam package`: utilice 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.
- `aws-sam deploy`: utilice este comando para implementar su 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 [Instale la CLI AWS SAM](#).

Plantillas de política de AWS SAM

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.

Temas

- [El AWS SAM proyecto y la AWS SAM plantilla](#)
- [Referencia de los comandos AWS SAM CLI](#)
- [Archivo de configuración de la CLI de AWS SAM](#)
- [AWS SAM referencia de conector](#)
- [Plantillas de política de AWS SAM](#)
- [Repositorios de imágenes para AWS SAM](#)
- [Telemetría en la AWS SAMCLI](#)
- [Configura y administra el acceso a los recursos en tu AWS SAM plantilla](#)

Referencia de los comandos AWS SAM CLI

Esta sección incluye información de referencia sobre AWS SAMCLI los comandos. Incluye 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 argumentos, variables de entorno y eventos. Consulte cada comando para obtener más información. Para obtener instrucciones sobre cómo instalar el AWS SAMCLI, consulte [Instale la CLI AWS SAM](#).

Temas

- [sam build](#)
- [sam delete](#)
- [sam deploy](#)
- [sam init](#)
- [sam list](#)
- [sam local generate-event](#)
- [sam local invoke](#)
- [sam local start-api](#)
- [sam local start-lambda](#)
- [sam logs](#)
- [sam package](#)
- [sam pipeline bootstrap](#)
- [sam pipeline init](#)
- [sam publish](#)
- [sam remote invoke](#)
- [sam remote test-event](#)
- [sam sync](#)
- [sam traces](#)
- [sam validate](#)

sam build

Esta página proporciona información de referencia para el AWS Serverless Application Model comando Command Line Interface (AWS SAMCLI) `sam build`.

- Para obtener una introducción al AWS SAMCLI, consulte [¿Qué es el? AWS SAMCLI](#).
- 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.

Uso

```
$ sam build <arguments> <options>
```

Argumentos

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. Utilice 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. Utilice esta opción para extraer la imagen desde otra ubicación.

Puede especificar esta opción varias veces. Cada instancia de esta opción puede tomar una cadena o un par clave-valor. Si especifica 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 especifica 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`

Proporcione `--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 los artefactos de compilación que no hayan cambiado con respecto a compilaciones anteriores. AWS SAM evalúa si ha cambiado algún archivo del directorio del 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 no 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, consulte [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, consulte [Archivo de configuración de la CLI de AWS SAM](#).

`--container-env-var`, `-e` *TEXT*

Las variables de entorno que transferir al contenedor de compilaciones. Puede 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, consulte [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 su 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.

--parallel

Compilaciones paralelas habilitadas. Utilice esta opción para crear las funciones y capas de la 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 AWS CloudFormation parámetros codificadas como pares clave-valor. Usa el mismo formato que (). AWS Command Line Interface AWS CLI Por ejemplo: 'ParameterKey=KeyName, ParameterValue=MyKey ParameterKey=InstanceType, ParameterValue=t1.micro'. Esta opción no es compatible con --hook-name.

--profile *TEXT*

El perfil específico del archivo de credenciales que recibe AWS las credenciales.

--region *TEXT*

El en el Región de AWS que se va a implementar. Por ejemplo, us-east-1.

--save-params

Guarde los parámetros que proporcione 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. Ejecute 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-container`, `-u`

Si las funciones dependen de paquetes que tienen dependencias compiladas de forma nativa, utilice esta opción para crear la función dentro de un contenedor de Docker tipo Lambda.

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 SAMCLI](#).

El `sam delete` comando elimina una AWS SAM aplicación eliminando la AWS 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.

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, consulte [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, consulte [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 AWS CloudFormation pila que desea eliminar.

sam deploy

Esta página proporciona información de referencia para el AWS Serverless Application Model comando Command Line Interface (AWS SAMCLI) `sam deploy`.

- Para obtener una introducción al AWS SAMCLI, consulte [¿Qué es el? AWS SAMCLI](#).
- Para obtener documentación sobre el uso del AWS SAMCLI `sam deploy` comando, consulte [Introducción a la implementación con AWS SAM](#).

El `sam deploy` comando despliega una aplicación para el usuario Nube de AWS . AWS CloudFormation

Uso

```
$ <environment variables> sam deploy <options>
```

Variables de entorno

SAM_CLI_POLL_DELAY

Establezca la variable de `SAM_CLI_POLL_DELAY` entorno con un valor de segundos para configurar la frecuencia con la que la CLI de AWS SAM comprueba el estado de la AWS CloudFormation pila, lo cual es útil cuando se ve la aceleración desde. AWS CloudFormation Esta variable `env` se utiliza para sondear las llamadas a la `describe_stack` API, que se realizan durante la ejecución. `sam deploy`

A continuación se muestra un ejemplo de esta variable:

```
$ SAM_CLI_POLL_DELAY=5 sam deploy
```

Opciones

`--capabilities` *LIST*

Una lista de capacidades que debe especificar para AWS CloudFormation permitir la creación de determinadas pilas. Algunas plantillas de pila pueden incluir recursos que afectan a sus permisos, por ejemplo Cuenta de AWS, mediante la creación de nuevos usuarios AWS Identity and Access Management (de 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`.

`--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, consulte [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, consulte [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`

Especifique si desea revertir la AWS 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 AWS CloudFormation pila vuelve al último estado estable. Si especifica `--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.

`--fail-on-empty-changeset` | `--no-fail-on-empty-changeset`

Especifique 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`

Especifique 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`

Especifique 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) que se utiliza para cifrar los artefactos que se encuentran en reposo en el bucket de Amazon S3. Si no especifica esta opción, utilizará claves de cifrado administradas por Amazon S3. AWS SAM

--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. Especifique esta opción si desea ver los cambios de la pila antes de aplicar el conjunto de cambios. Este comando crea un conjunto de cambios AWS CloudFormation y, a continuación, se cierra sin aplicar el conjunto de cambios. Para aplicar el conjunto de cambios, ejecute 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 de temas de Amazon Simple Notification Service (Amazon SNS) asociados a la pila AWS CloudFormation .


`--on-failure [ROLLBACK | DELETE | DO_NOTHING]`

Especifique 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

Especifique la opción `--disable-rollback` o la `--on-failure`, pero no ambas.

`--parameter-overrides`

Una cadena que contiene anulaciones de AWS CloudFormation parámetros codificadas como pares clave-valor. Usa el mismo formato que (). AWS Command Line Interface AWS CLI Por ejemplo, `ParameterKey=ParameterValue InstanceType=t1.micro`.

`--profile TEXT`


El perfil específico del archivo de credenciales que recibe AWS las credenciales.

`--region TEXT`

El en el Región de AWS que se va a implementar. Por ejemplo, `us-east-1`.

`--resolve-image-repos`

Cree 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 especifica la opción `--guided`, entonces la CLI de AWS SAM ignora `--resolve-image-repos`.

 Note

Si crea AWS SAM automáticamente cualquier repositorio de Amazon ECR para funciones o capas con esta opción y, posteriormente, elimina esas funciones o capas de la

AWS SAM plantilla, los repositorios de Amazon ECR correspondientes se eliminarán automáticamente.

`--resolve-s3`

Cree automáticamente un bucket de Amazon S3 para utilizarlo en el empaquetado y la implementación de implementaciones no guiadas. Si especifica la opción `--guided`, entonces la CLI AWS SAM ignora `--resolve-s3`. Si especifica 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 se AWS CloudFormation asume al aplicar el conjunto de cambios.

`--s3-bucket` *TEXT*

El nombre del bucket de Amazon S3 en el que este comando carga la AWS CloudFormation plantilla. Si la plantilla tiene más de 51 200 bytes, es necesaria la opción `--s3-bucket` o la opción `--resolve-s3`. Si especifica las opciones `--s3-bucket` y `--resolve-s3`, entonces se produce 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 proporcione 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 AWS CloudFormation pila en la que se va a realizar la implementación. Si especifica una pila existente, el comando la actualiza. Si especifica una nueva pila, el comando la crea.

`--tags` *LIST*

Una lista de etiquetas para asociarlas a la pila que se crea o actualiza. AWS 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-json`

Genera JSON para la AWS CloudFormation plantilla. El resultado predeterminado es YAML.

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 SAMCLI](#).
- 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.

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 Lambda de la aplicación. Especifique una de las siguientes opciones: `x86_64` o `arm64`

`--base-image` [*amazon/dotnet8-base* | *amazon/dotnet6-base* | *amazon/dotnetcore3.1-base* | *amazon/go1.x-base* | *amazon/java21-base* | *amazon/java17-base* | *amazon/java11-base* | *amazon/java8.al2-base* | *amazon/java8-base* | *amazon/nodejs20.x-base* | *amazon/nodejs18.x-base* | *amazon/nodejs16.x-base* | *amazon/python3.12-base* | *amazon/python3.11-base* | *amazon/python3.10-base* | *amazon/python3.9-base* | *amazon/python3.8-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, consulte [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, consulte [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-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 | dotnetcore3.1 | go1.x | java21 | java17 | java11 | java8 | java8.al2 | nodejs20.x | nodejs18.x | nodejs16.x | python3.12 | python3.11 | python3.10 | python3.9 | python3.8 | 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 AWS SAM de configuración.

`--tracing` | `--no-tracing`

Active el AWS X-Ray rastreo para sus funciones Lambda.

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 SAMCLI](#).

El comando `sam list` genera información importante sobre los recursos de la aplicación sin servidor y el estado de la aplicación sin servidor. Use `sam list` antes y después de la implementación para ayudar durante el desarrollo local y en la nube.

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 AWS 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 AWS CloudFormation pila a partir de una AWS 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 de la interfaz de línea de comandos (AWS SAMCLI).

Para obtener una introducción al AWS SAMCLI, consulte [¿Qué es el? AWS SAMCLI](#).

El `sam list endpoints` subcomando muestra una lista de puntos finales locales y en la nube de su AWS 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.

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, consulte [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, consulte [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]`

Especifique 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 AWS 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 AWS 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": "uj80uoe2o2",
    "CloudEndpoint": [
      "https://uj80uoe2o2.execute-api.us-east-1.amazonaws.com/Prod",
      "https://uj80uoe2o2.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 de la interfaz de línea de comandos (AWS SAMCLI).

Para obtener una introducción al AWS SAMCLI, consulte [¿Qué es el? AWS SAMCLI](#).

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.

Úselo `sam list resources` con una AWS SAM plantilla antes de la implementación para ver los recursos que se crearán. Proporcione un nombre de AWS 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.

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, consulte [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, consulte [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`]

Especifique 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 AWS 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 proporcionen, los ID lógicos de los recursos de la plantilla se asignarán a sus correspondientes ID físicos en AWS CloudFormation. Para obtener más información sobre los ID físicos, consulte los [Campos de recursos](#) en la Guía del usuario de AWS CloudFormation .

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 AWS 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 de la interfaz de línea de comandos (AWS SAMCLI).

Para obtener una introducción al AWS SAMCLI, consulte [¿Qué es el? AWS SAMCLI](#).

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 AWS CloudFormation plantilla.

Para obtener más información sobre Outputs, consulte [Salidas](#) en la Guía del usuario de AWS CloudFormation .

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, consulte [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, consulte [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]

Especifique 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 AWS 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 AWS CloudFormation `pilatest-stack`.

```
$ sam list stack-outputs --stack-name test-stack --output table
```

```
-----  
OutputKey                               OutputValue  
  Description  
-----
```

HelloWorldFunctionIamRole Implicit IAM Role created for Hello function	arn:aws:iam:: <i>account-number</i> :role/sam- app-test-list-HelloWorldFunctionRole- SRJDMJ6F7F41	World
HelloWorldApi Gateway endpoint URL for Prod for Hello World function	https://uj80uoe2o2.execute-api.us- east-1.amazonaws.com/Prod/hello/	API stage
HelloWorldFunction World Lambda Function ARN	arn:aws:lambda:us- east-1: <i>account-number</i> :function:sam-app- test-list- HelloWorldFunction-H85Y7yIV7ZLq	Hello

sam local generate-event

Esta página proporciona información de referencia para el AWS Serverless Application Model subcomando de la interfaz de línea de comandos (AWS SAMCLI). `sam local generate-event`

- Para obtener una introducción al AWS SAMCLI, consulte [¿Qué es el? AWS SAMCLI](#).
- Para obtener documentación sobre el uso del AWS SAMCLI `sam local generate-event` comando, consulte [Introducción a las pruebas con sam local generate-event](#).

El subcomando `sam local generate-event` genera muestras de carga útil de eventos para los Servicios de AWS compatibles.

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, consulte [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, consulte [Archivo de configuración de la CLI de AWS SAM](#).

`--help`

Muestra este mensaje y sale.

Servicio

Para ver una lista de los servicios compatibles, ejecute lo siguiente:

```
$ sam local generate-event
```

Evento

Para ver una lista de los eventos compatibles que se pueden generar para cada servicio, ejecute lo siguiente:

```
$ sam local generate-event <service>
```

Opciones de eventos

Para ver una lista de las opciones de eventos compatibles que puede modificar, ejecute lo siguiente:

```
$ sam local generate-event <service> <event> --help
```

sam local invoke

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 SAMCLI](#).
- Para obtener documentación sobre el uso del AWS SAMCLI `sam local invoke` subcomando, consulte [Introducción a las pruebas con sam local invoke](#).

El `aws sam local invoke` subcomando inicia una invocación única de una función de forma local.
AWS Lambda

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

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*

Transfiere una asignación de nombre de host a dirección IP al archivo de host del contenedor 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, consulte [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, consulte [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. Utilice 0.0.0.0 para enlazarse a todas las interfaces.

`--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.

`--env-vars, -n` *PATH*

Archivo JSON que contiene valores para las variables de entorno de la función de Lambda. Para obtener más información sobre los archivos de variable de entorno, consulte [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 AWS Lambda para 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 quiere usar para la invocación de la función local. De forma predeterminada, AWS SAM extrae la imagen del contenedor de Amazon ECR Public (que aparece en [Repositorios de imágenes para AWS SAM](#)). Utilice esta opción para extraer la imagen desde otra ubicación.

Por ejemplo, `sam local invoke MyFunction --invoke-image amazon/aws-sam-cli-emulation-image-python3.8`.

`--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.

`--no-event`

Invoca la función con un evento vacío.

`--parameter-overrides`

(Opcional) Una cadena que contiene anulaciones de AWS 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 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. Ejecute 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.

Especifique 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.

--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`.

sam local start-api

Esta página proporciona información de referencia para el AWS Serverless Application Model `sam local start-api` subcomando Command Line Interface (AWS SAMCLI).

- Para obtener una introducción al AWS SAMCLI, consulte [¿Qué es el? AWS SAMCLI](#).
- 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 AWS Lambda las funciones de forma local para probarlas a través de un servidor HTTP local.

Uso

```
$ sam local start-api <options>
```

Opciones

`--add-host` *LIST*

Transfiere una asignación de nombre de host a dirección IP al archivo de host del contenedor 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, consulte [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, consulte [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`. Utilice `0.0.0.0` para enlazarse a todas las interfaces.

`--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 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*

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*

Archivo JSON que contiene valores para las variables de entorno de la función de Lambda.

--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. De forma predeterminada, AWS SAM extrae la imagen del contenedor de Amazon ECR Public. Utilice esta opción para extraer la imagen desde otra ubicación.

Puede especificar esta opción varias veces. Cada instancia de esta opción puede tomar una cadena o un par clave-valor. Si especifica 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-python3.8`. Si especifica 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-python3.8 --invoke-image Function1=amazon/aws-sam-cli-emulation-image-python3.8`. 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.

--parameter-overrides

Opcional. Cadena que contiene anulaciones de AWS CloudFormation parámetros codificadas como pares clave-valor. Utilice el mismo formato que, AWS CLI por ejemplo, 'ParameterKey=, ParameterValue = KeyPairName MyKey ParameterKey, =t1.micro'. InstanceType ParameterValue

--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 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.

--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. Úselo 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 debe usarse la --ssl-key-file opción.

--ssl-key-file *PATH*


Ruta al archivo de claves SSL (predeterminada: ninguna). Al usar esta opción, también debe usarse la --ssl-cert-file opción.

`--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 de AWS SAM 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.

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 SAMCLI.](#)
- 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.

Uso

```
$ sam local start-lambda <options>
```

Opciones

`--add-host` *LIST*

Transfiere una asignación de nombre de host a dirección IP al archivo de host del contenedor 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, consulte [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, consulte [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`. Utilice `0.0.0.0` para enlazarse a todas las interfaces.

`--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 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*

Archivo JSON que contiene valores para las variables de entorno de la función de Lambda.

--force-image-build

Especifique 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. De forma predeterminada, AWS SAM extrae la imagen del contenedor de Amazon ECR Public. Utilice esta opción para extraer la imagen desde otra ubicación.

Por ejemplo, `sam local start-lambda MyFunction --invoke-image amazon/aws-sam-cli-emulation-image-python3.8`.

--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.

--parameter-overrides

Opcional. Cadena que contiene anulaciones de AWS CloudFormation parámetros codificadas como pares clave-valor. Utilice el mismo formato que, AWS CLI por ejemplo, 'ParameterKey=, ParameterValue = KeyPairName MyKey ParameterKey, =t1.micro'. InstanceType ParameterValue Esta opción no es compatible con `--hook-name`.

--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 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.

`--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. Ejecute 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.

sam logs

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 SAMCLI](#).

El `sam logs` comando busca los registros generados por sus AWS Lambda funciones.

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, consulte [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, consulte [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 los seguimientos de X-Ray 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 AWS CloudFormation pila, puede ser el ID lógico del recurso de la función en la AWS SAM plantilla AWS 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, especifique `text`. Para imprimir los registros como JSON, especifique `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 AWS 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 AWS 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
```

Utilice 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
```

Obtenga 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
```

Obtenga 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 [sam build](#) comando, ubicado en la `.aws-sam` subcarpeta y nombrado `template.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.

Uso

```
$ sam package <arguments> <options>
```

Argumentos

ID de recurso

El ID de la función 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, proporcione 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, consulte [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, consulte [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 claves de cifrado gestionadas por Amazon S3.

`--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 especifica una ruta, el comando escribe la plantilla en la salida estándar.

--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.

--resolve-s3

Creación automática de un bucket de Amazon S3 para empaquetar. Si especifica las opciones `--s3-bucket` y `--resolve-s3`, entonces se producirá un error.

--s3-bucket *TEXT*

El nombre del depósito de Amazon S3 en el que este comando carga el artefacto. Si el artefacto tiene más de 51.200 bytes, es necesaria la opción `--s3-bucket` o la `--resolve-s3` otra. Si especifica 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 de AWS SAM 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 AWS CloudFormation plantilla. YAML se utiliza de forma predeterminada.

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 SAMCLI](#).

El `sam pipeline bootstrap` subcomando genera los recursos de AWS infraestructura necesarios para conectarse a su sistema de CI/CD. 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 infraestructura: AWS

- Opción de configurar los permisos de canalización mediante:
 - Un usuario de IAM de canalización con un ID de clave de acceso y unas credenciales de acceso con clave secreta para compartirlas con el sistema CI/CD.

Note

Recomendamos rotar periódicamente las claves de acceso de forma regular. Para obtener más información, consulte [Rotar las claves de acceso periódicamente para casos de uso que requieran credenciales de larga duración](#) en la Guía del usuario de IAM.

- Plataformas CI/CD 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](#).
- Una función AWS CloudFormation de IAM de ejecución que asume AWS CloudFormation para implementar la AWS SAM aplicación.
- 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).

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/repository/admin/addon/admin/pipelines/openid-connect>

`--bucket` *TEXT*

El ARN del depósito de Amazon S3 que contiene los AWS SAM artefactos.

`--cicd-provider` *TEXT*

La plataforma de CI/CD para la canalización. AWS SAM

`--cloudformation-execution-role` *TEXT*

El ARN de la función de IAM que debe asumir AWS CloudFormation al implementar la pila de aplicaciones. Indíquelo 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, consulte [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, consulte [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`

Especifique si quiere crear un repositorio de imágenes de Amazon ECR si no se proporciona ninguno. En el repositorio de Amazon ECR se encuentran las imágenes de contenedor de las funciones o capas de Lambda 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 proveedor de CI/CD que se utilizará para los permisos de OIDC. GitLab, y GitHub 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 su ID de clave de acceso y su clave de acceso secreta con el sistema CI/CD. Se utiliza para conceder a este usuario de IAM permiso 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.

sam pipeline init

Esta página proporciona información de referencia para el AWS Serverless Application Model `sam local pipeline init` subcomando de la Interfaz de línea de comandos (AWS SAMCLI).

Para obtener una introducción al AWS SAMCLI, consulte [¿Qué es el? AWS SAMCLI](#).

El `sam pipeline init` subcomando genera un archivo de configuración de canalización que su sistema de CI/CD 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 consulte los recursos que haya creado previamente con el comando `sam pipeline bootstrap`.

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, consulte [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, consulte [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.

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 SAMCLI](#).

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 `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 [Publicar su solicitud con el AWS SAMCLI](#).

De forma predeterminada, `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, consulte [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, `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.

Uso

```
$ 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, consulte [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, consulte [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) Utilice 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, consulte 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 invoke

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 SAMCLI](#).
- Para obtener documentación sobre el uso del AWS SAMCLI `sam remote invoke` comando, consulte [Introducción a las pruebas en la nube con sam remote invoke](#).

El comando `sam remote invoke` invoca los recursos compatibles en Nube de AWS.

Uso

```
$ sam remote invoke <arguments> <options>
```

Argumentos

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.

i Tip

Use `aws 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 AWS CloudFormation.

i Tip

Use `aws sam list resources --stack-name <stack-name>` para obtener el ID físico de sus recursos.

Cuando proporciona un ARN o una identificación física:

Si proporciona un ARN o un ID físico, no proporcione 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 AWS 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 AWS 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 único 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 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*

Especifique el entorno que se va a utilizar en el archivo de configuración de la CLI de AWS SAM.

Valor predeterminado: default

`--config-file` *FILENAME*

Especifique la ruta y el nombre del archivo de configuración.

Para obtener más información sobre los archivos de configuración, consulte [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, consulte lo siguiente:

- Lambda con respuesta almacenada en búfer: [Lambda.Client.invoke](#)
- Lambda con transmisión de respuestas: [Lambda.Client.invoke_with_response_stream](#)

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 [SQS.Client.send_message](#).

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 de su 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 AWS CloudFormation pila a la que pertenece el recurso.

`--test-event-name NAME`

El nombre del evento de prueba compartible y pasarlo a la función de Lambda.

Note

Esta opción solo admite funciones de Lambda.

sam remote test-event

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 SAMCLI](#).
- Para obtener documentación sobre el uso del AWS SAMCLI `sam remote test-event` comando, consulte [Introducción a las pruebas en la nube con sam remote test-event](#).

El `sam remote test-event` comando interactúa con los eventos de prueba que se pueden compartir en el registro de EventBridge esquemas de Amazon.

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, consulte [sam remote test-event delete](#).

get

Obtenga un evento de prueba que se pueda compartir del registro de EventBridge esquemas. Para obtener más información de referencia, consulte [sam remote test-event get](#).

list

Enumere los eventos de prueba existentes que se pueden compartir para una AWS Lambda función. Para obtener más información de referencia, consulte [sam remote test-event list](#).

put

Guarde un evento de un archivo local en el registro de EventBridge esquemas. Para obtener más información de referencia, consulte [sam remote test-event put](#).

sam remote test-event delete

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 SAMCLI](#).
- Para obtener documentación sobre el uso del AWS SAMCLI `sam remote test-event` comando, consulte [Introducción a las pruebas en la nube con sam remote test-event](#).

El `sam remote test-event delete` subcomando elimina un evento de prueba que se puede compartir del registro de esquemas de Amazon EventBridge .

Uso

```
$ sam remote test-event delete <arguments> <options>
```

Argumentos

ID de recurso

El ID de la AWS Lambda función 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, consulte [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, consulte [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 se va a eliminar.

`--stack-name` *TEXT*

El nombre de la AWS 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 SAMCLI](#).
- Para obtener documentación sobre el uso del AWS SAMCLI `sam remote test-event` comando, consulte [Introducción a las pruebas en la nube con sam remote test-event](#).

El `sam remote test-event get` subcomando obtiene un evento de prueba que se puede compartir del registro de EventBridge esquemas de Amazon.

Uso

```
$ sam remote test-event get <arguments> <options>
```

Argumentos

ID de recurso

El ID de la AWS Lambda función 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, consulte [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, consulte [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 AWS 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 list

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 SAMCLI](#).
- Para obtener documentación sobre el uso del AWS SAMCLI `sam remote test-event` comando, consulte [Introducción a las pruebas en la nube con sam remote test-event](#).

El `sam remote test-event list` subcomando muestra los eventos de prueba que se pueden compartir existentes para una AWS Lambda función específica del registro de EventBridge esquemas de Amazon.

Uso

```
$ sam remote test-event list <arguments> <options>
```

Argumentos

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, consulte [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, consulte [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 AWS 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 put

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 SAMCLI](#).
- Para obtener documentación sobre el uso del AWS SAMCLI `sam remote test-event` comando, consulte [Introducción a las pruebas en la nube con sam remote test-event](#).

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.

Uso

```
$ sam remote test-event put <arguments> <options>
```

Argumentos

ID de recurso

El ID de la AWS Lambda función 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, consulte [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, consulte [Archivo de configuración de la CLI de AWS SAM](#).

`--file` *FILENAME*

La ruta del archivo y el nombre del evento en su máquina local.

Proporcione - como valor del nombre del archivo a leer de stdin.

Esta opción es obligatoria.

`--force`, `-f`

Sobrescriba un evento de prueba compartible 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 AWS CloudFormation pila asociada a la función Lambda.

Esta opción es obligatoria si proporciona el ID lógico de la función de Lambda como argumento.

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 SAMCLI](#).
- Para obtener documentación sobre el uso de AWS SAMCLI, consulte [El AWS SAMCLI](#).

El comando `sam sync` sincroniza los cambios de la aplicación local con Nube de AWS.

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. Utilice 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](#). Especifique 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: especifique 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
```

- Par clave-valor: especifique el nombre del recurso como clave y el URI de la imagen del contenedor que se utilizará con ese recurso como valor. Utilice 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 especifique para AWS 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). Especifique esta opción para anular los valores predeterminados. Entre los valores válidos se incluyen los siguientes:

- `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. Especifique 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 directamente. AWS CloudFormation Para actualizar tu AWS 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, consulte [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, consulte [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`

`--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 especifica esta opción, utilizará claves de cifrado administradas por Amazon S3. 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 AWS CloudFormation .

`--parameter-overrides`

Una cadena que contiene anulaciones de AWS CloudFormation parámetros codificadas como pares clave-valor. Use el mismo formato que AWS Command Line Interface (AWS CLI). Por ejemplo, `ParameterKey=ParameterValue InstanceType=t1.micro`.

`--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 AWS 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 AWS CloudFormation la plantilla. Si la plantilla tiene más de 51 200 bytes, se requiere la opción `--s3-bucket` o la `--resolve-s3`. Si especifica 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 SAMCLI comparará su AWS SAM plantilla local con la AWS CloudFormation plantilla desplegada y realizará una implementación solo si se detecta un cambio.

`--no-skip-deploy-sync` Especifica que se debe realizar un AWS CloudFormation despliegue cada vez `sam sync` que se ejecute.

Para obtener más información, consulte [Omita la AWS CloudFormation implementación inicial](#).

Valor predeterminado: `--skip-deploy-sync`

`--stack-name` *TEXT*

El nombre de la AWS 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. AWS 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 usa `--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 AWS CloudFormation despliegue inicial. A continuación, AWS SAM utiliza las API AWS de servicio para actualizar los recursos de código. AWS SAM AWS 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.

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 SAMCLI](#).

El `sam traces` comando busca AWS X-Ray rastros Cuenta de AWS en su interior. Región de AWS

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, consulte [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, consulte [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, especifique `text`. Para imprimir los registros como JSON, especifique `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

Ejecute el siguiente comando para obtener los rastros de X-Ray por ID.

```
$ sam traces --trace-id tracing-id-1 --trace-id tracing-id-2
```

Ejecute 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 SAMCLI](#).

El `sam validate` comando comprueba si un archivo AWS SAM de plantilla es válido.

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, consulte [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, consulte [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`

Ejecute la validación del elemento lint en la plantilla a través de cfn-lint. Cree un archivo de configuración `cfnlintrc` para especificar parámetros adicionales. Para obtener más información, consulte [cfn-lint en el repositorio](#).AWS 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.

AWS SAMCLIgestión

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 AWS SAMCLI comando mediante un archivo de configuración a nivel de proyecto. También incluye información sobre la administración de diferentes versiones de su dispositivo AWS SAMCLI, la configuración de AWS credenciales para AWS SAM poder realizar llamadas a AWS los servicios en su nombre y las diferentes formas de personalizarlas. 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 la CLI de AWS SAM](#)
- [Configuración de las credenciales de AWS](#)
- [Telemetría en la AWS SAMCLI](#)
- [Solución de problemas para AWS SAM CLI](#)

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, consulte [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, consulte [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
```

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`

Parámetros

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:
    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 la CLI de AWS SAM

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 desea, puede descargar e instalar la versión nocturna de la CLI de AWS SAM.

Temas

- [Actualización de la CLI de AWS SAM](#)
- [Desinstalación de la CLI de AWS SAM](#)
- [Cambie de usar Homebrew para administrar la CLI de AWS SAM](#)
- [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 la CLI de AWS SAM

Linux

Para actualizar la CLI de AWS SAM en Linux, siga las instrucciones de instalación en [Instalación de la CLI de AWS SAM](#), pero añada la opción `--update` al comando `install` de la siguiente manera:

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

macOS

AWS SAMCLI debe 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 CLI de AWS SAM mediante el instalador de paquetes, instale la última versión del paquete. Para ver instrucciones, consulte [Instalación de la CLI de AWS SAM](#).

Windows

Para actualizarla AWS SAMCLI, repita de [Instale la CLI de AWS SAM](#) nuevo los pasos de instalación de Windows.

Desinstalación de la CLI de AWS SAM

Linux

Para desinstalar la CLI de AWS SAM en Linux, debe eliminar el enlace simbólico y el directorio de instalación ejecutando los siguientes comandos:

1. Busque 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
```

- Busque 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. Elimine el enlace simbólico.

```
sudo rm /usr/local/bin/sam
```

3. Elimine el directorio de instalación.

```
sudo rm -rf /usr/local/aws-sam-cli
```

macOS

Desinstale la CLI de AWS SAM con el mismo método que utilizó para instalarla. Le recomendamos que utilice el instalador del paquete para instalar el AWS SAMCLI.

Si instaló la CLI de AWS SAM mediante el instalador del paquete, siga estos pasos para desinstalarlo.

Para desinstalar la CLI de AWS SAM

1. Elimine la CLI de AWS SAM 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 instalada la CLI de AWS SAM, sudo no es obligatorio. De lo contrario, se requiere el sudo.
 - b. **/path-to**: ruta al lugar donde instaló la CLI de AWS SAM. 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, sudo no es obligatorio. De lo contrario, se requiere el sudo.
 - b. **path-to-symlink-directory**— Su variable de \$PATH entorno. La ubicación predeterminada es /usr/local/bin.
3. Compruebe que la CLI de AWS SAM está desinstalada ejecutando lo siguiente:

```
$ sam --version
```

```
command not found: sam
```

Windows

Para desinstalar la CLI de AWS SAM mediante la configuración de Windows, siga estos pasos:

1. En el menú Inicio, busque “Agregar o quitar programas”.
2. Elija el resultado denominado AWS SAM Interfaz de línea de comandos de y, a continuación, seleccione Desinstalar para iniciar el desinstalador.
3. Confirme que desea desinstalar el AWS SAMCLI.

Cambie de usar Homebrew para administrar la CLI de AWS SAM

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 de Homebrew instalada con AWS SAM](#) para desinstalar la versión Homebrew gestionada.
2. Siga las instrucciones que aparecen en [Instale la CLI AWS SAM](#) para instalar la CLI de AWS SAM mediante un método compatible.

Administrar la compilación nocturna de la CLI de AWS SAM

Puede descargar e instalar la compilación nocturna de la CLI de AWS SAM. Contiene una versión preliminar del código de la CLI de AWS SAM que puede ser menos estable que la versión de producción. Una vez instalada, puede usar la compilación nocturna con el comando `sam-nightly`. Puede 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, siga estas instrucciones:

Linux

Puede instalar la versión de compilación nocturna de la CLI de AWS SAM en la plataforma Linux x86_64 mediante el instalador de paquetes.

Para instalar la compilación nocturna de la CLI de AWS SAM

1. Descargue el instalador del paquete desde [sam-cli-nightly](#) el aws-sam-cli GitHub repositorio.
2. [Siga los pasos para instalar el paquete de compilación nocturna de la CLI de AWS SAM.](#)

macOS

Puede instalar la versión de compilación nocturna de la CLI de AWS SAM en macOS mediante el instalador del paquete de compilación nocturna.

Para instalar la compilación nocturna de la CLI de AWS SAM

1. Descargue el instalador del paquete para su plataforma desde [sam-cli-nightly](#) el aws-sam-cli GitHub repositorio.
2. [Siga 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, siga los mismos pasos que en [Instale la CLI AWS SAM](#), pero utilice en su lugar el enlace de descarga de la compilación nocturna.

Para comprobar que ha instalado la versión de compilación nocturna, ejecute el comando `sam-nightly --version`. El resultado de este comando tiene el formato `1.X.Y.dev<YYYYMMDDHHmm>`, por ejemplo:

```
SAM CLI, version 1.20.0.dev202103151200
```

Cambie del Homebrew al instalador del paquete

Si va a instalar Homebrew y actualizar la compilación nocturna de la CLI de AWS SAM y desea pasar a usar el instalador de paquetes, siga estos pasos.

Para cambiar del Homebrew al instalador de paquetes

1. Desinstale la Homebrew instalada de la CLI de AWS SAM.

```
$ brew uninstall aws-sam-cli-nightly
```

2. Compruebe 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 debe 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, consulte [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 elija, cree un entorno virtual y asígnele un nombre.

Linux / macOS

```
$ mkdir project  
$ cd project
```

```
$ python3 -m venv venv
```

Windows

```
> mkdir project  
> cd project  
> py -3 -m venv venv
```

2. Active 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. Instale la CLI de AWS SAM en el entorno virtual.

```
(venv) $ pip install --upgrade aws-sam-cli
```

4. Compruebe que la CLI de AWS SAM esté instalada correctamente.

```
(venv) $ sam --version  
SAM CLI, version 1.94.0
```

5. Puede utilizar el comando `deactivate` para salir del entorno virtual. Siempre que inicie una sesión nueva, debe 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, puede 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`.
Le recomendamos que utilice 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, consulte el repositorio [homebrew-core](#).

Para instalar la CLI de AWS SAM

1. Ejecute 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, ejecute el siguiente comando:

```
$ brew upgrade aws-sam-cli
```

Desinstalar una CLI de Homebrew instalada con AWS SAM

Si la CLI de AWS SAM se instaló utilizando Homebrew, siga estos pasos para desinstalarla.

Para desinstalar la CLI de AWS SAM

1. Ejecute lo siguiente:

```
$ brew uninstall aws-sam-cli
```

2. Compruebe que la CLI de AWS SAM 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, ejecute lo siguiente:

```
$ brew uninstall aws-sam-cli && brew untap aws/tap && brew cleanup aws/tap && brew  
update && brew install aws-sam-cli
```

Siga 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. Compruebe que la CLI de AWS SAM se haya desinstalado:

```
$ which sam
```

```
sam not found
```

- 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
```

- Elimine los archivos en caché del instalador AWS gestionado:

```
$ brew cleanup aws/tap
```

- Actualización de Homebrew y todas las fórmulas:

```
$ brew update
```

- Instale la versión gestionada por la comunidad de AWS SAMCLI:

```
$ brew install aws-sam-cli
```

- Compruebe 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 [Solución de problemas para AWS SAM CLI](#).

Configuración de las credenciales de AWS

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 AWS CloudFormation.

Es posible que ya haya establecido AWS las credenciales para trabajar con AWS herramientas, como uno de los AWS SDK o el 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 más información acerca de las claves de acceso, consulte [Administración de las claves de acceso para usuarios de IAM](#) en la Guía del usuario de IAM.

A continuación, determine si la tiene AWS CLI instalada. A continuación, siga las instrucciones en una de las siguientes secciones:

Uso de AWS CLI

Si la tienes AWS CLI instalada, usa el `aws configure` comando y sigue 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`, consulte [Configuración rápida de la AWS CLI](#) en la Guía del usuario de AWS Command Line Interface .

Sin utilizar la 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: puede configurar las variables de entorno `AWS_ACCESS_KEY_ID` y `AWS_SECRET_ACCESS_KEY`.

Para configurar estas variables en Linux o macOS, utilice 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, use 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 SAM CLI

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, consulte [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, ejecute 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

Ejecute los siguientes comandos para desactivar la telemetría en todas las sesiones cuando ejecute AWS SAM CLI en el sistema operativo.

Para desactivar la telemetría en Linux

1. Ejecute:

```
echo "export SAM_CLI_TELEMETRY=0" >> ~/.profile
```

2. Ejecute:

```
source ~/.profile
```

Para desactivar la telemetría en macOS

1. Ejecute:

```
echo "export SAM_CLI_TELEMETRY=0" >> ~/.profile
```

2. Ejecute:

```
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 usa 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 la terminal se cierre y se vuelva a abrir.

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 más información, consulte los siguientes temas:

- [AWS Términos del servicio](#)
- [Preguntas frecuentes sobre privacidad de datos](#)

Solución de problemas para AWS SAM CLI

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 de](#) .

Mensajes de error

Error de Curl: “curl: (6) No se pudo resolver:...”

Al intentar invocar el punto final de API Gateway, aparece 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 implementado correctamente mediante la AWS CloudFormation consola o el AWS CLI, y compruebe que el curl comando es correcto.

Error: no se puede encontrar la información exacta del recurso con el nombre de pila dado

Al ejecutar el comando `sam remote invoke` en una aplicación que contiene un único recurso de función 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 no ARN se proporciona una función como argumento, el `sam remote invoke` comando requiere que se proporcione la `--stack-name` opción.

Solución: proporcione 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 `sam remote invoke` comando y pasar una función Lambda ARN 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, pase el ID lógico de la función.

Puede pasar el identificador lógico de la función como argumento en lugar del de la función. ARN

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 pase la AWS SAMCLI función ARN 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 de AWS servicio. Para solucionar este problema, debe configurar AWS las credenciales. Para obtener más información, consulte [Configuración de las credenciales de AWS](#).

Error: FileNotFoundError en Windows

Al ejecutar comandos AWS SAMCLI en Windows, es posible que aparezca el siguiente error:

```
Error: FileNotFoundError
```

Causa posible: AWS SAMCLI podría interactuar con las rutas de archivo que superen el límite máximo de rutas de Windows.

Solución: para resolver este problema, es necesario activar el nuevo comportamiento de las rutas largas. Para ello, consulte [Habilitar rutas largas en Windows 10, versión 1607 y versiones posteriores](#), en la documentación de desarrollo de aplicaciones de 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 el AWS SAMCLI, se producirá el error del solucionador de dependencias.

Soluciones:

1. Utilice el instalador de paquetes nativo AWS SAMCLI.
 - a. Desinstalación de la AWS SAM CLI con pip. Para ver instrucciones, consulte [Desinstalación de la CLI de AWS SAM](#).
 - b. Instale AWS SAMCLI mediante el instalador de paquetes nativo. Para ver instrucciones, consulte [Instale la CLI de AWS SAM](#).
 - c. Cuando sea necesario, AWS SAMCLI actualice con el instalador de paquetes nativo. Para ver instrucciones, consulte [Actualización de la CLI de AWS SAM](#).
2. Si debe usar pip, le recomendamos que lo instale AWS SAM CLI en un entorno virtual. Esto garantiza un entorno de instalación limpio y un entorno aislado en caso de que se produzcan errores. Para ver instrucciones, consulte [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 ver instrucciones, consulte [Actualización de la CLI de AWS SAM](#).

Error: la ejecución de AWS SAM proyectos localmente requiere Docker. ¿Lo tiene 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, consulte [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 un Amazon API Gateway de acceso público API configurado sin autorización. Al responder **N** a esta pregunta, está diciendo que esto no está bien.

Para resolver este problema, dispone de las siguientes opciones:

- Configure su aplicación con autorización. Para obtener más información acerca de la autorización, consulte [Controla el API acceso con tu AWS SAM plantilla](#).
- Si su intención es disponer de un API punto final 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 final de API Gateway, aparece el siguiente error:

```
{"message": "Missing Authentication Token"}
```

Esto significa que has intentado enviar una solicitud al dominio correcto, pero URI no es reconocible. Para solucionar este problema, verifica el comando completo URL y actualiza el curl comando con el correctoURL.

Mensajes de advertencia

Advertencia:... AWS dejará de mantener el Homebrew instalador durante 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 mantiene el Homebrew soporte.

A partir de septiembre de 2023, AWS dejará de mantener el Homebrew instalador del AWS SAMCLI.

Solución: utilice un método de instalación AWS compatible.

- Encontrará los métodos de instalación AWS compatibles en [Instale la CLI AWS SAM](#).

Solución: para seguir utilizando Homebrew, utilice el instalador gestionado por la comunidad.

- Puede utilizar el instalador Homebrew gestionado por la comunidad a su entera discreción. Para obtener instrucciones, consulte [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 [Para obtener más información, consulte administración de los permisos de recursos con conectores de AWS SAM ..](#)

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::Connector](#).

Para los recursos de origen y destino, cuando se definan en la misma plantilla, utilice la propiedad `Id`. Si lo desea, puede añadirse un `Qualifier` para reducir el alcance del recurso definido. Cuando el recurso no esté dentro de la misma plantilla, utilice 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
<code>AWS::ApiGateway::RestApi</code>	<code>AWS::Lambda::Function</code>	Write	<code>Id</code> o <code>Qualifier</code> , <code>ResourceId</code> y <code>Type</code>	<code>Id</code> o <code>Arn</code> y <code>Type</code>
<code>AWS::ApiGateway::RestApi</code>	<code>AWS::Serverless::Function</code>	Write	<code>Id</code> o <code>Qualifier</code> , <code>ResourceId</code> y <code>Type</code>	<code>Id</code> o <code>Arn</code> y <code>Type</code>
<code>AWS::ApiGatewayV2::Api</code>	<code>AWS::Lambda::Function</code>	Write	<code>Id</code> o <code>Qualifier</code> , <code>ResourceId</code> y <code>Type</code>	<code>Id</code> o <code>Arn</code> y <code>Type</code>
<code>AWS::ApiGatewayV2::Api</code>	<code>AWS::Serverless::Function</code>	Write	<code>Id</code> o <code>Qualifier</code> , <code>ResourceId</code> y <code>Type</code>	<code>Id</code> o <code>Arn</code> y <code>Type</code>
<code>AWS::AppSync::DataSource</code>	<code>AWS::DynamoDB::Table</code>	Read	<code>Id</code> o <code>RoleName</code> y <code>Type</code>	<code>Id</code> o <code>Arn</code> y <code>Type</code>
<code>AWS::AppSync::DataSource</code>	<code>AWS::DynamoDB::Table</code>	Write	<code>Id</code> o <code>RoleName</code> y <code>Type</code>	<code>Id</code> o <code>Arn</code> y <code>Type</code>

Tipo de origen	Tipo de destino	Permisos	Propiedades de origen	Propiedades de destino
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
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

Tipo de origen	Tipo de destino	Permisos	Propiedades de origen	Propiedades de destino
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
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

Tipo de origen	Tipo de destino	Permisos	Propiedades de origen	Propiedades de destino
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
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

Tipo de origen	Tipo de destino	Permisos	Propiedades de origen	Propiedades de destino
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
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::EventBus	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::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
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

Tipo de origen	Tipo de destino	Permisos	Propiedades de origen	Propiedades de destino
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
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::EventBus	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

Tipo de origen	Tipo de destino	Permisos	Propiedades de origen	Propiedades de destino
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
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

Tipo de origen	Tipo de destino	Permisos	Propiedades de origen	Propiedades de destino
AWS::SNS: :Topic	AWS::Lamb da::Funct ion	Write	Id o Arn y Type	Id o Arn y Type
AWS::SNS: :Topic	AWS::Serv erless::F unction	Write	Id o Arn y Type	Id o Arn y Type
AWS::SNS: :Topic	AWS::SQS: :Queue	Write	Id o Arn y Type	Id o Arn, QueueUrl y Type
AWS::SQS: :Queue	AWS::Lamb da::Funct ion	Read, Write	Id o Arn y Type	Id o RoleName y Type
AWS::SQS: :Queue	AWS::Serv erless::F unction	Read, Write	Id o Arn y Type	Id o RoleName y Type
AWS::Step Functions ::StateMa chine	AWS::Dyna moDB::Tab le	Read, Write	Id o RoleName y Type	Id o Arn y Type
AWS::Step Functions ::StateMa chine	AWS::Even ts::Event Bus	Write	Id o RoleName y Type	Id o Arn y Type
AWS::Step Functions ::StateMa chine	AWS::Lamb da::Funct ion	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::S3::Bucket	Read, Write	Id o RoleName y Type	Id o Arn y Type
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 CLI de AWS SAM

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 de Docker](#)
- [Siguiendo pasos](#)

Instalación de 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 crear y ejecutar funciones de Lambda con una arquitectura de conjunto de instrucciones diferente a la de la máquina host, hay pasos adicionales para configurar 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 de](#) . O bien, consulte la sección de [Resolución de problemas](#) de los Pasos posteriores a la instalación en 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, esté incluido 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. Verifique 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 [Instale la CLI 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 la creación de imágenes de contenedores. Las imágenes que se AWS SAM proporcionan 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 utilizar estas imágenes con sistemas de CI/CD para automatizar la creación y el despliegue de aplicaciones. AWS SAM Para ver ejemplos, consulte [Implemente con sistemas y canalizaciones de CI/CD](#).

AWS SAM URIsLas imágenes del contenedor de compilación están etiquetadas con la versión AWS SAM CLI incluida en esa imagen. Si especificas la que no está etiquetada URI, se utilizará 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 AWS SAM CLI versión 1.24.1.

A partir de la versión 1.33.0 de AWS SAM CLI, ambas imágenes x86_64 y las del arm64 contenedor están disponibles para los tiempos de ejecución compatibles. Para obtener más información, consulte [Tiempos de ejecución de Lambda](#) en la Guía para desarrolladores de AWS Lambda .

Note

Antes de la versión 1.22.0 AWS SAM CLI, DockerHub era el repositorio predeterminado del que AWS SAM CLI extraía la imagen del contenedor. A partir de la versión 1.22.0, el repositorio predeterminado pasó a ser 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 [sam build](#) con la opción `--build-image`. Los ejemplos que aparecen al final de este tema muestran cómo crear aplicaciones utilizando imágenes de DockerHub repositorios.

Repositorio de imágenes URIs

En la siguiente tabla se enumeran las imágenes URIs de contenedores de compilación de [Amazon ECR Public](#) que puede utilizar para crear y empaquetar aplicaciones sin AWS SAM servidor.

Note

Amazon ECR Public lo reemplazó DockerHub a partir de la AWS SAMCLI versión 1.22.0. 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-provid.al2023
Tiempo de ejecución personalizado (AL2)	public.ecr.aws/sam/build-provided.al2
Tiempo de ejecución personalizado	public.ecr.aws/sam/build-provided
Go 1.x	public.ecr.aws/sam/build-go1.x
Java 21	public.ecr.aws/sam/build-java21
Java 17	public.ecr.aws/sam/build-java17
Java 11	public.ecr.aws/sam/build-java11
Java 8 (AL2)	public.ecr.aws/sam/build-java8.al2
Java 8	public.ecr.aws/sam/build-java8
.NET8	public.ecr.aws/sam/build-dotnet8
.NET7	public.ecr.aws/sam/build-dotnet7

Tiempo de ejecución	Amazon ECR Public
.NET6	public.ecr.aws/sam/build-dotnet6
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 3.12	public.ecr.aws/sam/build-python3.12
Python 3.11	public.ecr.aws/sam/build-python3.11
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 3.3	public.ecr.aws/sam/build-ruby3.3
Ruby 3.2	public.ecr.aws/sam/build-ruby3.2

Ejemplos

Los dos comandos de ejemplo siguientes crean aplicaciones utilizando imágenes de contenedores del DockerHub repositorio:

Cree una aplicación Node.js 20 utilizando una imagen de contenedor extraída de DockerHub:

```
$ sam build --use-container --build-image public.ecr.aws/sam/build-nodejs20.x
```

Cree un recurso de función utilizando la imagen del contenedor Python 3.12 extraída de DockerHub:

```
$ sam build --use-container --build-image Function1=public.ecr.aws/sam/build-python3.12
```

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 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, puede 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.
- Si se activan las CloudWatch alarmas, revierte automáticamente el despliegue.

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 AWS Management Console.

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: nodejs12.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 está implementando código nuevo, en función de los cambios en el Amazon S3 de la función Lambda. URI
 - 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 está familiarizado con el control de versiones y los alias de las funciones de Lambda, consulte [Control de versiones y alias de las funciones de AWS Lambda](#).
- **Deployment Preference Type:** En el ejemplo anterior, el 10 por ciento 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, revierte la CodeDeploy implementación. Puede especificar cómo debe desviarse el tráfico entre versiones de las siguientes maneras:
 - **Canary:** el tráfico se desvía en dos incrementos. Puede 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. Puede 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
Canary10Percent5Minutes
Canary10Percent10Minutes
Canary10Percent15Minutes
Lineal: 10 (10 minutos) PercentEvery
Lineal PercentEvery 10:1 minuto
Lineal: 10PercentEvery, 2 minutos
Lineal: 10PercentEvery, 3 minutos
AllAtOnce

- **Alarms:** Se trata de CloudWatch alarmas que se activan por cualquier 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:** funciones de prueba 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 que se ha realizado correctamente o se ha producido un error. Si la función falla, se interrumpe e informa del error a. AWS 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.

Utilice enlaces posteriores al tráfico para ejecutar pruebas de integración u otras acciones de validación.

Para obtener más información, consulte [SAMReference to Safe Deployments](#).

Implementación gradual de una función 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: implemente la función de Lambda y cree alias automáticamente con `AutoPublishAlias`.
- Paso 2: realice su 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: implementación de la función de Lambda

```
Resources:
MyLambdaFunction:
  Type: AWS::Serverless::Function
  Properties:
    Handler: index.handler
    Runtime: nodejs12.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: nodejs12.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, consulte 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 2020](#)

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 AWS SAM CLI instala `pip` el
- Si usa `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, consulte [Política de obsolescencia del tiempo de ejecución](#) en la Guía del desarrollador de AWS Lambda .

Notas importantes para 2020

Junio de 2020

Instalación de la CLI de AWS SAM en Windows 32 bits

El soporte para la CLI de AWS SAM en Windows de 32 bits pronto quedará obsoleto. Si utiliza un sistema de 32 bits, le recomendamos que lo actualice a uno de 64 bits y siga las instrucciones que se encuentran en [Instale la CLI AWS SAM](#).

Si no puede actualizar a un sistema de 64 bits, puede utilizar la [caja de herramientas Legacy Docker](#) de la CLI de AWS SAM en un sistema de 32 bits. Sin embargo, esto provocará que encuentre ciertas limitaciones con el AWS SAMCLI. Por ejemplo, no puede ejecutar contenedores Docker de 64 bits en un sistema de 32 bits. Por lo tanto, si su función de Lambda depende de un contenedor de 64 bits compilado de forma nativa, no podrá probarla localmente en un sistema de 32 bits.

Para instalar la CLI de AWS SAM en un sistema de 32 bits, ejecute el siguiente comando:

```
pip install aws-sam-cli
```

Important

Aunque el `pip install aws-sam-cli` comando también funciona en Windows de 64 bits, le recomendamos que utilice el sistema de [64 bits MSI](#) para instalarlo AWS SAMCLI en sistemas de 64 bits.

Ejemplos de aplicaciones sin servidor para AWS SAM

En esta sección se incluyen dos aplicaciones de ejemplo: 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 las fuentes y AWS los recursos de los eventos. 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 Lambda invocada por un origen 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

Asegúrese de que ha realizado la configuración requerida en el [Instale la CLI AWS SAM](#).

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. Ejecute el siguiente comando en un indicador de comandos CLI de la AWS SAM.

```
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. Revise el contenido del directorio creado por el comando (`dynamodb_event_reader/`):
 - `template.yaml`— Defina 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, utilice 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.

Compruebe 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. Cree un bucket de S3 en la ubicación donde desea guardar el código empaquetado. Si desea utilizar un bucket de S3 existente, omita este paso.

```
aws s3 mb s3://bucketname
```

2. Cree el paquete de despliegue ejecutando el siguiente `package` CLI comando en la línea 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 utilizará para desplegar la aplicación AWS en la nube. A continuación, pruebe la aplicación.

Para implementar la aplicación sin servidor en la nube AWS

- En el AWS SAMCLI, utilice el `deploy` CLI comando 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 `--capabilities` parámetro AWS CloudFormation permite crear un IAM rol.

AWS CloudFormation crea los AWS recursos que se definen en la plantilla. Puede acceder a los nombres de estos recursos en la AWS CloudFormation consola.

Para probar la aplicación sin servidor en la nube AWS

1. Abra 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

El AWS SAM GitHub repositorio contiene aplicaciones de ejemplo adicionales para descargar y experimentar con ellas. Para acceder a este repositorio, consulte los [Ejemplos de aplicaciones de AWS SAM](#).

Procese eventos de Amazon S3 con AWS SAM

Con esta aplicación de ejemplo, puede basarse en lo aprendido en los ejemplos anteriores e instalar una aplicación más compleja. Esta aplicación consiste en una función de Lambda invocada por un origen de eventos de subida 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 sin 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 se IAM configuren los permisos antes de poder probar la función Lambda localmente. Vamos a aprovechar AWS CloudFormation para crear los recursos y configurar los permisos por usted. De lo contrario, tendrá que hacerlo manualmente antes de poder probar la función de Lambda localmente.

Como este ejemplo es más complicado, asegúrese de estar familiarizado con la instalación de las aplicaciones del ejemplo anterior antes de ejecutar esta.

Antes de empezar

Asegúrese de que ha realizado la configuración requerida en el [Instale la CLI AWS SAM](#).

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. Ejecute el siguiente comando en un indicador de comandos CLI de la AWS SAM.

```
sam init \  
--location https://github.com/aws-samples/cookiecutter-aws-sam-s3-rekognition-  
dynamodb-python \  
--no-input
```

2. Revise el contenido del directorio creado por el comando (`aws_sam_ocr/`):
 - `template.yaml`— Defina 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 localmente.

Creación del paquete de implementación de Lambda

1. Cree un bucket de S3 en la ubicación donde desea guardar el código empaquetado. Si desea utilizar un bucket de S3 existente, omita este paso.

```
aws s3 mb s3://bucketname
```

2. Cree el paquete de implementación ejecutando el siguiente package CLI comando en la línea 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: implementación de la aplicación

Ahora que ha creado el paquete de despliegue, lo utilizará para desplegar la aplicación AWS en la nube. A continuación, pruebe la aplicación invocándola en la AWS nube.

Para implementar la aplicación sin servidor en la nube AWS

- 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 AWS CloudFormation permite crear un IAM rol.

AWS CloudFormation crea los AWS recursos que se definen en la plantilla. Puede acceder a los nombres de estos recursos en la AWS CloudFormation consola.

Para probar la aplicación sin servidor en la nube AWS

1. Cargue una imagen en el bucket de Amazon S3 que creó para esta aplicación de muestra.
2. Abra la consola de DynamoDB y busque la tabla que se creó. Consulte la tabla para ver los resultados devueltos por Amazon Rekognition.
3. Compruebe que la tabla de DynamoDB contenga nuevos registros que contengan texto que Amazon Rekognition encontró en la imagen cargada.

Paso 4: probar la aplicación localmente

Antes de poder probar la aplicación localmente, primero debe recuperar los nombres de los AWS recursos creados por AWS CloudFormation.

- Recupere el nombre de la clave y el nombre del bucket de Amazon S3 AWS CloudFormation. Modifique el `SampleEvent.json` archivo sustituyendo los valores de la clave de objeto, el nombre del depósito y el depósitoARN.
- Recupere el nombre de la tabla de DynamoDB Este nombre se utiliza para el siguiente comando `sam local invoke`.

Utilice 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 AWS 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 puede comprobar que se crearon los registros de DynamoDB esperados, en función de los resultados devueltos por Amazon Rekognition.

Siguientes pasos

El AWS SAM GitHub repositorio contiene aplicaciones de ejemplo adicionales para descargar y experimentar con ellas. Para acceder a este repositorio, consulte los [Ejemplos de aplicaciones de AWS SAM](#).

Compatibilidad de la CLI de AWS SAM con Terraform

En esta sección se describe el uso de la interfaz de línea de AWS Serverless Application Model comandos (AWS SAMCLI) con tus Terraform proyectos y con Terraform la nube.

Para enviar comentarios y solicitudes de características, cree un [Problema de GitHub](#).

Temas

- [Introducción a compatibilidad de Terraform con CLI de AWS SAM](#)
- [Uso de la CLI de AWS SAM con Terraform para la depuración y las pruebas locales](#)
- [Uso de la CLI de AWS SAM con Serverless.tf para la depuración y las pruebas locales](#)
- [CLI de AWS SAM con referencia de Terraform](#)
- [¿Cuál es la compatibilidad de la CLI de AWS SAM con Terraform?](#)

Introducción a compatibilidad de Terraform con CLI de AWS SAM

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

- [Requisitos previos de Terraform para la CLI de AWS SAM](#)
- [Uso de comandos de la CLI de AWS SAM con Terraform](#)
- [Configuración para proyectos de Terraform](#)
- [Configuración para Terraform Cloud](#)

Requisitos previos de Terraform para la CLI de AWS SAM

Complete todos los requisitos previos para empezar a utilizar la CLI de AWS SAM en sus proyectos de Terraform.

1. Instale o actualice la CLI de AWS SAM

Para comprobar si tiene la CLI de AWS SAM instalada, ejecute 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, consulte [Actualización de la CLI de AWS SAM](#).

Para obtener instrucciones sobre la instalación de la CLI de AWS SAM junto con todos sus requisitos previos, consulte [Instale la CLI de AWS SAM](#).

2. Instalar Terraform

Para comprobar si instaló Terraform, ejecute lo siguiente:

```
$ terraform -version
```

Para instalar Terraform, consulte [Instalar Terraform](#) en el registro de Terraform.

3. Instale Docker para realizar pruebas locales

La CLI de AWS SAM requiere Docker para las pruebas locales. Para instalar Docker, consulte [Instalación de Docker para usarlo con la CLI de AWS SAM](#).

Uso de comandos de la CLI de AWS SAM con Terraform

Cuando ejecute un comando compatible con la CLI de AWS SAM, utilice la opción `--hook-name` y proporcione el valor `terraform`. A continuación, se muestra un ejemplo:

```
$ sam local invoke --hook-name terraform
```

Puede configurar esta opción en su archivo de configuración de la CLI de AWS SAM con lo siguiente:

```
hook_name = "terraform"
```

Configuración para proyectos de Terraform

Complete los pasos de este tema para utilizar la CLI de AWS SAM con proyectos de Terraform.

No se requiere ninguna configuración adicional si construyes tus AWS Lambda artefactos fuera del Terraform proyecto. Consulte [Uso de la CLI de AWS SAM con Terraform para la depuración y las pruebas locales](#) para empezar a utilizar el AWS SAM CLI.

Si crea los artefactos de Lambda en sus proyectos de Terraform, debe hacer lo siguiente:

1. Instale Python 3.8 o una versión más reciente
2. Instale la herramienta Make.
3. Defina la lógica de creación de artefactos de Lambda en su proyecto de Terraform.
4. Defina 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.

Instale Python 3.8 o una versión más reciente

PythonSe requiere la versión 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, consulte la página [Downloading Python](#) (Descarga de Python) en la Guía para principiantes de Python.

Compruebe que Python 3.8 o una versión posterior se haya agregado a la ruta de su máquina ejecutando:

```
$ python --version
```

El resultado debe mostrar una versión de Python 3.8 o posterior.

Instale la herramienta Make.

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 ha instalado Make en su máquina local, instálelo antes de continuar.

Para Windows, puede realizar la instalación mediante [Chocolatey](#). Para obtener instrucciones, consulte [Uso de Chocolatey](#) en Cómo instalar y usar “Make” en Windows

Defina la lógica de construcción de los artefactos de Lambda

Utilice 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 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 recurso sam metadata

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, consulte [null_resource](#) en el registro de Terraform.

Para definir un recurso sam metadata

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 su recurso.
3. Especifique 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 sam metadata recurso variará según el tipo de recurso de Lambda (función o capa) y el tipo de empaquetado (ZIPo imagen). Para obtener más información y ejemplos, consulte [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 que haya generado este archivo, puede 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 API puntos de conexión.

Utilice la CLI de AWS SAM para crear sus artefactos de 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 sam metadata en su proyecto de Terraform para conocer y localizar sus 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 `sam local` comandos.

Para crear con `sam build`

1. Desde el directorio que contiene el módulo raíz de Terraform, ejecute lo siguiente:

```
$ sam build --hook-name terraform
```

2. Para crear una función o capa de Lambda específica, ejecute lo siguiente

```
$ 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, debe especificar la ubicación. Utilice 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:

```
$ sam build --hook-name terraform --terraform-project-root-path ~/projects/terraform/demo
```

Creación mediante un contenedor

Al ejecutar el AWS SAMCLI `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 ver instrucciones, consulte [Instalación de Docker para usarlo con la CLI de AWS SAM](#).

Para compilar usando un contenedor

1. Cree un `Dockerfile` que contenga las herramientas Terraform, Python y Make. También debe 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
```

Siguientes pasos

Para empezar a utilizar la CLI de AWS SAM en sus proyectos de Terraform, consulte [Uso de la CLI de AWS SAM con Terraform para la depuración y las pruebas locales](#).

Configuración para Terraform Cloud

Se recomienda usar Terraform v1.6.0 o posterior. Si usa una versión anterior, debe generar un archivo de plan de Terraform de forma local. El archivo del plan local AWS SAM CLI proporciona la información necesaria para realizar las pruebas y la depuración locales.

Para generar un archivo de plan local

Note

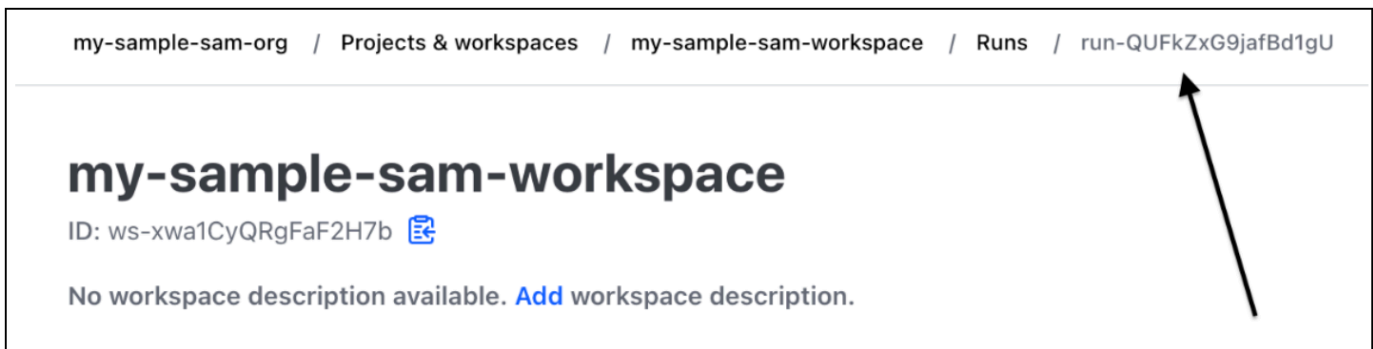
Estos pasos no son necesarios con Terraform v1.6.0 o posterior. Para empezar a utilizar el AWS SAM CLI with Terraform Cloud, consulte [Uso de la CLI de AWS SAM con Terraform](#).

1. Configurar un API token: el tipo de token dependerá de su nivel de acceso. Para obtener más información, consulta [API los tokens](#) en la Terraform Cloud documentación.
2. Configura la variable de entorno del API token: el siguiente es un ejemplo de la línea de comandos:

```
$ export TOKEN="<api-token-value>"
```

3. Obtén tu ID de ejecución: en la Terraform Cloud consola, busca el ID de Terraform ejecución de la ejecución que quieres usar con AWS SAMCLI.

El ID de ejecución se encuentra en la ruta de navegación del proceso de ejecución.



4. Obtén el archivo del plan: con tu API token, obtén 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 puede utilizar la CLI de AWS SAM con Terraform Cloud. Si utiliza un comando compatible con la CLI de AWS SAM, utilice 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, consulte [api_gateway_v2_tf_cloud](#) en el repositorio de GitHub `aws-samples`.

Siguientes pasos

Para comenzar a usar la CLI de AWS SAM con Terraform Cloud, consulte [Uso de la CLI de AWS SAM con Terraform para la depuración y las pruebas locales](#).

Uso de la CLI de AWS SAM con Terraform para la depuración y las 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 sam local invoke](#)
- [Pruebas locales con sam local start-api](#)
- [Pruebas locales con sam local start-lambda](#)
- [Limitaciones de Terraform](#)

Pruebas locales con sam local invoke

Note

Para usar la CLI de AWS SAM para realizar pruebas locales, debe tener Docker instalado y configurado. Para ver instrucciones, consulte [Instalación de Docker para usarlo con la CLI de AWS SAM](#).

El siguiente es un ejemplo de prueba local de la función de Lambda 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, consulte [Introducción a las pruebas con sam local invoke](#).

Pruebas locales con sam local start-api

Para utilizar `sam local start-api` con Terraform, ejecute 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, consulte [Introducción a las pruebas con sam local start-api](#).

Funciones 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 Lambda que utilizan autorizadores de Lambda](#).
- Para obtener más información sobre el uso de autorizadores Lambda en Terraform, consulte [Resource: aws_api_gateway_authorizer](#) en el registro de Terraform.

Pruebas locales con sam local start-lambda

El siguiente es un ejemplo de cómo probar la función Lambda localmente con AWS Command Line Interface (AWS CLI):

1. Utilice la CLI de AWS SAM para crear un entorno de pruebas local:

```
$ sam local start-lambda --hook-name terraform hello_world_function
```

2. Usa el AWS CLI para invocar tu función localmente:

```
$ 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, consulte [Introducción a las pruebas con sam local start-lambda](#).

Limitaciones de Terraform

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 de cuerpo del REST API recurso.

Para evitar estas limitaciones, puede ejecutar `terraform apply` cuando se agrega un nuevo recurso.

Uso de la CLI de AWS SAM con Serverless.tf para la depuración y las pruebas locales

La interfaz de línea de AWS Serverless Application Model comandos (AWS SAMCLI) se puede usar con los módulos Serverless.tf para depurar y probar localmente las funciones y capas. AWS Lambda 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

Las versiones 4.6.0 y posteriores de Serverless.tf admiten la integración de la CLI de AWS SAM.

Para empezar a utilizarla AWS SAMCLI con sus módulos Serverless.tf, actualice a la última versión Serverless.tf y a la. AWS SAMCLI

A partir de la versión 6.0.0 de serverless.tf, debe establecer el parámetro `create_sam_metadata` como `true`. Esto genera los recursos de metadatos que requiere el comando. `AWS SAMCLI sam build`

Para obtener más información de `Serverless.tf`, consulte la [terraform-aws-lambda-module](#).

CLI de AWS SAM con referencia de Terraform

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:

- [sam build](#)
- [sam local invoke](#)
- [sam local start-api](#)
- [sam local start-lambda](#)

Referencia específica de Terraform

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 [¿Cuál es la compatibilidad de la CLI de AWS SAM con Terraform?](#).
- Para utilizar la CLI de AWS SAM con Terraform, consulte [Uso de la CLI de AWS SAM con Terraform para la depuración y las pruebas locales](#).

Temas

- [Argumentos](#)
- [Ejemplos](#)

Argumentos

Argumento	Descripción
<code>built_output_path</code>	La ruta a los artefactos creados por tu AWS Lambda función.
<code>docker_build_args</code>	Cadena decodificada del objeto Build Arguments JSON 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>	La ruta del archivo de Docker. Esta ruta es relativa a la ruta <code>docker_context</code> . Este argumento es opcional. El valor predeterminado es <code>Dockerfile</code> .
<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, consulte El argumento depends_on en el registro de Terraform.
<code>original_source_code</code>	La ruta a la que se define la función de Lambda. Este valor puede ser una cadena, una matriz de cadenas o un JSON objeto decodificado en forma de cadena. <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 JSON caso de los objetos, también se <code>source_code_property</code> debe definir.

Argumento	Descripción
<code>resource_name</code>	El nombre de la función de Lambda.
<code>resource_type</code>	El formato del tipo de paquete de funciones de Lambda. Los valores aceptados son: <ul style="list-style-type: none"> • <code>IMAGE_LAMBDA_FUNCTION</code> • <code>LAMBDA_LAYER</code> • <code>ZIP_LAMBDA_FUNCTION</code>
<code>source_code_property</code>	La ruta al código de recursos de Lambda en el JSON objeto. Defina esta propiedad cuando <code>original_source_code</code> sea un JSON objeto.

Ejemplos

sam: recurso de metadatos que hace referencia a una función Lambda mediante ZIP el tipo de paquete

```
# 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
]
}

```

recurso de metadatos de sam que hace referencia a una función de Lambda mediante el tipo de paquete de imagen

```

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]
}

```

¿Cuál es la compatibilidad de la CLI de AWS SAM con 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.
- Amazon API Gateway APIs.

Para ver una introducción a Terraform, consulte [¿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 AWS SAMCLI la integración, AWS SAMCLI los procesos y los comandos del usuario 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 SAMCLI](#)
- [¿Cómo puedo usar la CLI de AWS SAM con Terraform?](#)
- [Siguiendo pasos](#)

¿Qué es el? AWS SAMCLI

AWS SAMCLISE trata de una herramienta de línea de comandos que puede utilizar con AWS SAM plantillas e integraciones de terceros compatibles, por ejemplo Terraform, para crear y ejecutar aplicaciones sin servidor. Para obtener una introducción a la AWS SAMCLI, consulte [¿Qué es el? AWS SAMCLI](#)

The AWS SAMCLI admite los siguientes comandos para Terraform:

- `aws 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 `aws sam local invoke`](#).
- `aws sam local start-api`— Ejecute sus recursos de Lambda localmente y pruébelos a través de un HTTP servidor local. Este tipo de pruebas son útiles para las funciones Lambda que invoca un punto final de API Gateway. Para obtener más información acerca de este comando, consulte [Introducción a las pruebas con `aws sam local start-api`](#).
- `aws sam local start-lambda`— Inicie un punto final local para su función Lambda para invocarla localmente mediante AWS Command Line Interface (AWS CLI) o SDKs Para obtener más

información acerca de este comando, consulte [Introducción a las pruebas con sam local start-lambda](#).

¿Cómo puedo usar la CLI de AWS SAM con 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 gestionar sus aplicaciones. AWS En general, esto significa lo siguiente:

- Escribir: cree su infraestructura como código mediante Terraform.
- Probar y depurar: use la CLI de AWS SAM para probar y depurar sus aplicaciones de forma local.
- Planificar: obtenga 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.

Siguientes pasos

Para completar todos los requisitos previos y configurar Terraform, consulte [Introducción a compatibilidad de Terraform con CLI de AWS SAM](#).

Pruebe y cree AWS CDK aplicaciones a nivel local con AWS SAMCLI

Puede usar la CLI de AWS SAM para probar y crear localmente aplicaciones sin servidor definidas mediante AWS Cloud Development Kit (AWS CDK). Como AWS SAMCLI funciona dentro de la estructura del AWS CDK proyecto, puede seguir utilizando el [AWS CDK kit de herramientas](#) para crear, modificar e implementar sus AWS CDK aplicaciones.

Para obtener información sobre la instalación y la configuración del AWS CDK, consulte [Introducción a él AWS CDK en la](#) Guía para AWS Cloud Development Kit (AWS CDK) desarrolladores.

Note

AWS SAMCLIEs compatible con la versión AWS CDK v1 a partir de la versión 1.135.0 y la AWS CDK v2 a partir de la versión 2.0.0.

Temas

- [Empezar con AWS SAM y el AWS CDK](#)
- [Prueba local de AWS CDK aplicaciones con AWS SAM](#)
- [Creación de AWS CDK aplicaciones con AWS SAM](#)
- [Implementación AWS CDK de aplicaciones en AWS SAM](#)

Empezar con AWS SAM y el AWS CDK

En este tema se describe lo que se necesita para usar AWS SAMCLI con las AWS CDK aplicaciones y se proporcionan instrucciones para crear y probar localmente una AWS CDK aplicación sencilla.

Requisitos previos

Para usar AWS SAMCLI with AWS CDK, debe instalar el AWS CDK, y el AWS SAMCLI.

- Para obtener información sobre cómo instalar el AWS CDK, consulte [Introducción a él AWS CDK](#) en la Guía para AWS Cloud Development Kit (AWS CDK) desarrolladores.
- Para obtener información sobre cómo instalar el AWS SAMCLI, consulte [Instale la CLI AWS SAM](#).

Crear y probar localmente una aplicación AWS CDK

Para probar localmente una AWS CDK aplicación mediante el AWS SAMCLI, debe tener una AWS CDK aplicación que contenga una función Lambda. Siga los pasos siguientes para crear una AWS CDK aplicación básica con una función Lambda. Para obtener más información, consulte [Crear una aplicación sin servidor usando la AWS CDK](#) en la Guía para desarrolladores de AWS Cloud Development Kit (AWS CDK) .

Note

AWS SAMCLI admite la versión AWS CDK v1 a partir de la versión 1.135.0 y la AWS CDK v2 a partir de la versión 2.0.0.

Paso 1: Crear una aplicación de AWS CDK

Para este tutorial, inicialice una AWS CDK aplicación que utilice TypeScript

Comando a ejecutar:

AWS CDK v2

```
mkdir cdk-sam-example
cd cdk-sam-example
cdk init app --language typescript
```

AWS CDK v1

```
mkdir cdk-sam-example
cd cdk-sam-example
cdk init app --language typescript
npm install @aws-cdk/aws-lambda
```

Paso 2: Añadir una función de Lambda a la aplicación

Reemplace el código en `lib/cdk-sam-example-stack.ts` con lo siguiente:

AWS CDK v2

```
import { Stack, StackProps } from 'aws-cdk-lib';
```

```
import { Construct } from 'constructs';
import * as lambda from 'aws-cdk-lib/aws-lambda';

export class CdkSamExampleStack extends Stack {
  constructor(scope: Construct, id: string, props?: StackProps) {
    super(scope, id, props);

    new lambda.Function(this, 'MyFunction', {
      runtime: lambda.Runtime.PYTHON_3_9,
      handler: 'app.lambda_handler',
      code: lambda.Code.fromAsset('./my_function'),
    });
  }
}
```

AWS CDK v1

```
import * as cdk from '@aws-cdk/core';
import * as lambda from '@aws-cdk/aws-lambda';

export class CdkSamExampleStack extends cdk.Stack {
  constructor(scope: Construct, id: string, props?: StackProps) {
    super(scope, id, props);

    new lambda.Function(this, 'MyFunction', {
      runtime: lambda.Runtime.PYTHON_3_9,
      handler: 'app.lambda_handler',
      code: lambda.Code.fromAsset('./my_function'),
    });
  }
}
```

Paso 3: Añadir el código de la función de Lambda

Cree un directorio llamado `my_function`. En dicho directorio, cree un archivo llamado `app.py`.

Comando a ejecutar:

```
mkdir my_function
cd my_function
touch app.py
```


Añada el siguiente código a `app.py`:

```
def lambda_handler(event, context):  
    return "Hello from SAM and the CDK!"
```

Paso 4: comprobación de la función de Lambda

Puede utilizarla para AWS SAMCLI invocar localmente una función Lambda que defina en AWS CDK una aplicación. Para ello, necesita el identificador de construcción de la función y la ruta a la plantilla sintetizada AWS CloudFormation .

Comando a ejecutar:

```
cdk synth --no-staging
```

```
sam local invoke MyFunction --no-event -t ./cdk.out/CdkSamExampleStack.template.json
```

Ejemplo del resultado:

```
Invoking app.lambda_handler (python3.9)
```

```
START RequestId: 5434c093-7182-4012-9b06-635011cac4f2 Version: $LATEST
```

```
"Hello from SAM and the CDK!"
```

```
END RequestId: 5434c093-7182-4012-9b06-635011cac4f2
```

```
REPORT RequestId: 5434c093-7182-4012-9b06-635011cac4f2 Init Duration: 0.32 ms Duration:  
177.47 ms Billed Duration: 178 ms Memory Size: 128 MB Max Memory Used: 128 MB
```

Para obtener más información sobre las opciones disponibles para probar AWS CDK aplicaciones mediante la AWS SAM CLI, consulte [Prueba local de AWS CDK aplicaciones con AWS SAM](#).

Prueba local de AWS CDK aplicaciones con AWS SAM

Puede usarlo AWS SAMCLI para probar sus AWS CDK aplicaciones de forma local ejecutando los siguientes comandos desde el directorio raíz del proyecto de su AWS CDK aplicación:

- [sam local invoke](#)
- [sam local start-api](#)
- [sam local start-lambda](#)

Antes de ejecutar cualquiera de los `awslocal` comandos con una AWS CDK aplicación, debe ejecutar `cdk synth`.

Al ejecutar `awslocal invoke`, necesitará el identificador de construcción de la función que desea invocar y la ruta a la AWS CloudFormation plantilla sintetizada. Si la aplicación utiliza pilas anidadas, para resolver conflictos de nomenclatura también necesitará el nombre de la pila en la que se define la función.

Uso:

```
# Invoke the function FUNCTION_IDENTIFIER declared in the stack STACK_NAME
awslocal invoke [OPTIONS] [STACK_NAME/FUNCTION_IDENTIFIER]

# Start all APIs declared in the AWS CDK application
awslocal start-api -t ./cdk.out/CdkSamExampleStack.template.json [OPTIONS]

# Start a local endpoint that emulates AWS Lambda
awslocal start-lambda -t ./cdk.out/CdkSamExampleStack.template.json [OPTIONS]
```

Ejemplo

Considerar las pilas y las funciones que se declaran en el siguiente ejemplo:

```
app = new HelloCdkStack(app, "HelloCdkStack",
    ...
)
class HelloCdkStack extends cdk.Stack {
    constructor(scope: Construct, id: string, props?: cdk.StackProps) {
        ...
        new lambda.Function(this, 'MyFunction', {
            ...
        });

        new HelloCdkNestedStack(this, 'HelloNestedStack' ,{
            ...
        });
    }
}

class HelloCdkNestedStack extends cdk.NestedStack {
    constructor(scope: Construct, id: string, props?: cdk.NestedStackProps) {
        ...
        new lambda.Function(this, 'MyFunction', {
```

```
    ...
  });
  new lambda.Function(this, 'MyNestedFunction', {
    ...
  });
}
```

Los siguientes comandos invocan localmente las funciones de Lambda definidas en el siguiente ejemplo:

```
# Invoke MyFunction from the HelloCdkStack
sam local invoke -t ./cdk.out/HelloCdkStack.template.json MyFunction
```

```
# Invoke MyNestedFunction from the HelloCdkNestedStack
sam local invoke -t ./cdk.out/HelloCdkStack.template.json MyNestedFunction
```

```
# Invoke MyFunction from the HelloCdkNestedStack
sam local invoke -t ./cdk.out/HelloCdkStack.template.json HelloNestedStack/MyFunction
```

Creación de AWS CDK aplicaciones con AWS SAM

La CLI de AWS SAM proporciona soporte para crear funciones y capas de Lambda definidas en su aplicación AWS CDK con [sam build](#).

En el caso de las funciones de Lambda que utilizan artefactos zip, ejecute `cdk synth` antes de ejecutar comandos `sam local`. `sam build` no es obligatorio.

Si AWS CDK la aplicación utiliza funciones del tipo de imagen, ejecute los `sam local` comandos `cdk synth` y, a continuación, `sam build` ejecútelos antes de ejecutar. Cuando se ejecuta `sam build`, AWS SAM no crea funciones Lambda ni capas que utilicen construcciones específicas del tiempo de ejecución, por ejemplo, [NodejsFunction](#) `sam build` [no admite activos agrupados](#).

Ejemplo

Al ejecutar el siguiente comando desde el directorio raíz AWS CDK del proyecto, se crea la aplicación.

```
sam build -t ./cdk.out/CdkSamExampleStack.template.json
```

Implementación AWS CDK de aplicaciones en AWS SAM

No AWS SAMCLI admite la implementación de AWS CDK aplicaciones. Use `cdk deploy` para implementar la aplicación. Para obtener más información, consulte [Kit de herramientas \(comando cdk\) de AWS CDK](#) en la Guía para desarrolladores de AWS Cloud Development Kit (AWS CDK) .

Publicar su solicitud con el AWS SAMCLI

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 debe haberla probado localmente o en AWS Cloud.

Siga 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, consulte [Propiedades de la sección de metadatos de la plantilla de AWS SAM](#).

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, consulte [Instale la CLI de AWS SAM](#). Para determinar si la CLI de AWS SAM está instalado en su clúster, ejecute 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 puede 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 su 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 su aplicación. Para configurar esta política, haga lo siguiente:
 1. Abra la consola de Amazon S3 en <https://console.aws.amazon.com/s3>.
 2. Elija el nombre del bucket de Amazon S3 que utilizó para empaquetar su aplicación.
 3. Elija Permisos.
 4. En la pestaña Permissions (Permisos), en Bucket policy (Política de bucket), elija Edit (Editar).
 5. En la página Editar política de buckets, 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, consulte [Referencia de los elementos de la política de IAM](#) en la Guía del usuario de IAM.

```
{
  "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. Elija Guardar cambios.

Publicación de una aplicación

Paso 1: Añadir una **Metadata** sección a la 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 [Propiedades de la sección de metadatos de la plantilla de AWS SAM](#).

Paso 2: crear el paquete de la aplicación

Ejecute 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, utilice 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 su aplicación en el AWS Serverless Application Repository. También puede ir directamente a la [AWS Serverless Application Repository página de inicio](#) y buscar su 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 su aplicación mediante el AWS CLI, consulte los [ejemplos de políticas AWS Serverless Application Repository basadas en recursos](#) en la Guía para AWS Serverless Application Repository desarrolladores. Para obtener información sobre cómo compartir su aplicación mediante la AWS Management Console, consulte [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 Lambda o haya agregado un nuevo componente a la arquitectura de la aplicación.

Para actualizar una aplicación que haya publicado anteriormente, vuelva a publicarla siguiendo el mismo proceso detallado anteriormente. En la sección Metadata del archivo de plantilla de AWS SAM, proporcione el mismo nombre de aplicación con el que la publicó originalmente, pero incluya un valor `SemanticVersion` nuevo.

Por ejemplo, considere 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

- [Propiedades de la sección de metadatos de la plantilla de AWS SAM](#)

Propiedades de la sección de metadatos de la plantilla de AWS SAM

`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

AWS 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 Metadatos 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, consulte 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 sam package comando puede tener una referencia a un archivo local para esta propiedad. Sin embargo, para que una aplicación</p>

Propiedad	Tipo	Obligatorio	Descripción
			<p>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> <p>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.</p>
<code>ReadmeUrl</code>	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>
<code>Labels</code>	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>

Propiedad	Tipo	Obligatorio	Descripción
HomePageUrl	Cadena	FALSO	Una URL con más información sobre la aplicación, por ejemplo, la ubicación del GitHub repositorio de la aplicación.
SemanticVersion	Cadena	FALSO	La versión semántica de la aplicación. Para obtener más información sobre el control de versiones semántico, consulte el sitio web Control de versiones semántico . Debe proporcionar un valor de esta propiedad para que la aplicación sea pública.
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 función 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, puede suscribirse a una fuente RSS.

- Última actualización de la documentación: 20 de junio de 2024

Cambio	Descripción	Fecha
Contenido reestructurado y actualizado en toda la guía para desarrolladores	Se reorganizó y reestructuró la guía para mejorar la visibilidad y la usabilidad. Títulos actualizados y mejorados. Proporcionó detalles adicionales al presentar los temas y conceptos.	20 de junio de 2024
Se agregó AWS SAMCLI soporte para Ruby 3.3	Ruby 3.3 ya está disponible como repositorio de imágenes y de tiempo de ejecución . Consulte Repositorios de imágenes y sam init para obtener más información.	4 de abril de 2024
Se agregaron opciones AWS SAMCLI de comando	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 localstart-api y sam local start-lambda	20 de marzo de 2024

[AWS SAMCLISe agregó soporte para .NET 8](#)

.NET 8 ya está disponible como un repositorio de imágenes y de tiempo de ejecución. Ya no se admiten los tiempos de ejecución y los repositorios de imágenes de .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 AWS SAMCLI arm64 para Linux](#)

Para obtener instrucciones, consulte [Instalación del AWS SAMCLI](#).

6 de diciembre de 2023

[Se agregó la opción --watch-exclude para el comando sam sync AWS SAMCLI](#)

Excluya archivos y carpetas de modo que no inicien una sincronización. Para obtener más información, consulte [Especificar archivos y carpetas que no iniciarán una sincronización](#).

6 de diciembre de 2023

[Se agregó la build-in-source opción -- para el comando sam sync AWS SAMCLI](#)

Compile el proyecto en la carpeta de origen para acelerar el proceso de compilación. Para obtener más información, consulte [Acelere los tiempos de compilación al compilar el proyecto en la carpeta de origen](#).

6 de diciembre de 2023

Se agregó la build-in-source opción -- para el comando AWS SAMCLI sam build	Compile el proyecto en la carpeta de origen para acelerar el proceso de compilación. Para obtener más información, consulte Acelere los tiempos de compilación al compilar el proyecto en la carpeta de origen .	6 de diciembre de 2023
Se agregó un nuevo soporte de recursos para el AWS SAMCLI comando de invocación remota	Use <code>sam remote invoke</code> con las aplicaciones de Kinesis Data Streams, las colas de Amazon SQS y las máquinas de estado de Step Functions. Para obtener más información, consulte Uso de la invocación remota de sam .	15 de noviembre de 2023
Se agregó un nuevo comando AWS SAMCLI <code>remote test-event</code> para eventos de prueba que se pueden compartir	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, consulte Uso del comando <code>sam remote test-event</code> .	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, consulte compatibilidad de la CLI de AWS SAM para Terraform .	5 de septiembre de 2023

Se agregó soporte AWS SAMCLI para Terraform Cloud	La CLI de AWS SAM ahora es compatible con las pruebas locales para Terraform Cloud. Para obtener más información, consulte Configurar para Terraform Cloud .	5 de septiembre de 2023
Se agregó soporte de formato de YAML archivo para el archivo AWS SAMCLI de configuración	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 SAMCLIsam local start-api comandos para Terraform	El ¿Para qué AWS SAMCLI sirve el soporteTerraform? la sección se ha actualizado para incluir AWS SAMCLIsam local start-api el soporte de comandos paraTerraform.	6 de julio de 2023
Se agregó un nuevo AWS SAMCLI comando de invocación remota	Para empezar a usar sam remote invoke, consulte Uso de invocación remota sam .	22 de junio de 2023
Se agregó un tipo de AWS AppSyncGraphQL API recurso sin servidor	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 AWS SAMCLI soporte para Ruby 3.2	Actualice la página sam init para incluir una nueva imagen base y valores de tiempo de ejecución. Actualice 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 AWS SAMCLI del paquete	Actualice la página Instalación de la CLI de AWS SAM para que refleje el paso opcional. Cree 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 Sam Sync para omitir la sincronización de la infraestructura	Personalice si se requiere una AWS CloudFormation implementación cada vez <code>sam sync</code> que se ejecute. Para obtener más información, consulte Omitir la AWS CloudFormation implementación inicial .	23 de marzo de 2023
Se agregó soporte para el tipo de fuente de eventos de DocumentDB	La especificación AWS SAM de la plantilla ahora admite el tipo de fuente de DocumentDB eventos para el <code>AWS::Serverless::Function</code> recurso. Para obtener más información, consulte DocumentDB .	10 de marzo de 2023

Creación de funciones de Lambda Rust con Cargo Lambda	Use la CLI de AWS SAM para crear sus funciones de Lambda Rust utilizando Cargo Lambda. Para obtener más información, consulte 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	Utilice la nueva sintaxis de conectores integrados para definir sus recursos <code>AWS::Serverless::Connector</code> . 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 <code>sam list</code> para la CLI de AWS SAM	Use <code>sam list</code> para ver información importante sobre los recursos de la aplicación sin servidor. Para obtener más información, consulte sam list .	2 de febrero de 2023

[Se agregaron propiedades de formato y OutExtension compilación para esbuild](#)

La creación de funciones de Lambda de Node.js con esbuild ahora admite propiedades de compilación `Format` y `OutExtension` . Para obtener más información, consulte [Creación de funciones de Lambda de Node.js con esbuild](#).

2 de febrero de 2023

[Se agregaron opciones de administración del tiempo de ejecución a la especificación de la AWS SAM plantilla](#)

Configure las opciones de administración del tiempo de ejecución para las funciones de Lambda. Para obtener más información, consulte [RuntimeManagementConfig](#) .

24 de enero de 2023

[Se agregó la propiedad EventSource de destino al AWS::Serverless::StateMachine recurso.](#)

El tipo de recurso `AWS::Serverless::StateMachine` admite la propiedad `Target` 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](#)

Configure el escalado de los sondeadores de SQS con la propiedad `ScalingConfig` para `AWS::Serverless::Function` . Para obtener más información, consulte [ScalingConfig](#) .

12 de enero de 2023

Valide AWS SAM las aplicaciones con cfn-lint	Puede usar cfn-lint para validar sus plantillas a través del. AWS SAM AWS SAMCLI Para obtener más información, consulte 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	Instale la AWS SAM de CLI con el nuevo instalador del paquete macOS. Para obtener más información, consulte Instalación del AWS SAMCLI .	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, consulte 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 sus aplicaciones sin servidor. Para obtener más información, consulte [Administrar el acceso y los permisos a los recursos](#).

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 anticipada AWS SAM(AOT) para mejorar los tiempos de arranque en frío de Lambda. Para obtener más información, consulte [Creación de funciones de Lambda de .NET 7 con la compilación AOT nativa](#).

15 de noviembre de 2022

[Se ha añadido compatibilidad de Terraform para la CLI de AWS SAM para la depuración y las pruebas locales](#)

Use la CLI de AWS SAM en sus proyectos de Terraform para realizar depuraciones y pruebas locales de sus funciones y capas de Lambda. Para obtener más información, consulte [Soporte de Terraform para la CLI de AWS SAM](#).

14 de noviembre de 2022

[EventBridge Se agregó soporte para Scheduler 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 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 SAM CLI](#).

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 pueden provocar errores al ejecutarse `sam init` debido a las limitaciones `MAX_PATH` de Windows 10. Para obtener más información, consulte [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, consulte 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	<code>FilterCriteria</code> 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 <code>issuer</code> y <code>audience</code> bajo <code>JwtConfiguration</code> 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 `--on-failure` agregada a [sam deploy](#).

9 de septiembre de 2022

La compatibilidad esbuild ya está disponible con carácter general	Para compilar y empaquetar las funciones Lambda de Node.js, 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	Utilice variables de entorno de la CLI de AWS SAM cuando invoque funciones de Lambda de forma local y cuando ejecute API Gateway de forma local .	1 de septiembre de 2022
Compatibilidad con arquitecturas del conjunto de instrucciones de Lambda	Use la CLI de AWS SAM 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 <code>AWS::Serverless::Function</code> recurso y la CompatibleArchitectures propiedad del tipo de recurso. <code>AWS::Serverless::LayerVersion</code>	1 de octubre de 2021

[Generación de ejemplos de configuraciones de canalización](#)

Utilice la CLI de AWS SAM para generar canalizaciones de ejemplo para varios sistemas de CI/CD mediante los comandos nuevos [sam pipeline bootstrap](#) y [sam pipeline init](#). Para obtener más información, consulte [Generación de canalizaciones de CI/CD 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, consulte [AWS Cloud Development Kit \(AWS CDK\) \(versión preliminar\)](#).

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, consulte 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 sistemas de CI/CD comunes AWS CodePipeline, como Jenkins, GitLab CI/CD y Actions. GitHub Para obtener más información, consulte [Implementación de aplicaciones sin servidor](#).

24 de junio de 2021

[AWS SAM AWS Depuración de aplicaciones con kits de herramientas](#)

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 SAM CLI para probar y crear AWS CDK aplicaciones de forma local. Esta es una versión preliminar pública. Para obtener más información, consulte [AWS Cloud Development Kit \(AWS CDK\) \(versión preliminar\)](#).

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, consulte 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 elija en Instalación del. AWS SAMCLI	25 de marzo de 2021
Compatibilidad con variables del entorno de compilación de contenedores	Ahora puede pasar variables de entorno para crear contenedores. Para obtener más información, consulte las opciones <code>--container-env-var</code> y <code>--container-env-var-file</code> en sam build .	4 de marzo de 2021
Nuevo proceso de instalación de Linux	Ahora puede instalar la CLI de AWS SAM con un instalador nativo de Linux. Para obtener más información, consulte 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, consulte 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, consulte 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, consulte 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 pruebas locales mediante los comandos [sam local start-api](#) y [sam local start-lambda](#) de la CLI de AWS SAM. Para obtener más información, consulte 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, consulte [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](#) y el 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, consulte 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, consulte [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, consulte [Archivo de configuración de la CLI de AWS SAM](#).

24 de septiembre de 2020

[Soporte para X-Ray con Step Functions y referencias para controlar el acceso a las API](#)

Se ha añadido soporte para X-Ray como origen de eventos para máquinas de estado sin servidor. Para obtener más información, consulte 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, consulte 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 su función de Lambda. Para obtener más información, consulte 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 su función de Lambda acceder a los recursos compartidos y también le permite modificarlos. Para obtener más información, consulte 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 <code>AWS::Serverless::StateMachine</code> 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, consulte acerca de la creación de un proveedor de identidad personalizado, consulte Creación de tiempos de ejecución personalizados .	21 de mayo de 2020
Creación de capas	Se ha añadido la capacidad de crear recursos <code>LayerVersion</code> individuales. Para obtener más información, consulte Creación de capas .	19 de mayo de 2020
Recursos generados AWS CloudFormation	Proporcionó detalles sobre los AWS CloudFormation recursos que AWS SAM genera y cómo hacer referencia a ellos. Para obtener más información, consulte AWS 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 [Configurar 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, consulte la [especificación de AWS SAM](#). También se actualizó el flujo de trabajo de implementación con cambios en el comando `aws 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, consulte [Control del acceso a las API de API Gateway](#). También se actualizaron dos plantillas de políticas: RekognitionFacesPolicy y ElasticsearchHttpPolicy. Para obtener más información, consulte [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, consulte [Control del acceso a las API de API Gateway](#).

21 de marzo de 2019

[Se ha añadido el `publish` a la CLI de AWS SAM](#)

El nuevo comando `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, consulte [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, consulte [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 SAM CLI. [Para obtener más información, consulte Instalación del AWS SAM CLI](#)

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 versión original de inglés, prevalecerá la versión en inglés.